GOVP1199217935

제 1 차 년 도
연 차 보 고 서

Visualization 소프트웨어 개발에 관한 연구(Ⅰ)

— A Study on the Development of Visualization Software(Ⅰ) —

연 구 기 관
한국과학기술연구원
시스템공학연구소

과 학 기 술 처

배 포 선

사 본 번 호	부수	배 포 처
1/100	1	시스템공학연구소 영구보존용
2/100	1	시스템공학연구소 도서과 보관용
3/100 - 6/100	4	시스템공학연구소 연구관리과 보관용
7/100 - 8/100	2	시스템공학연구소 CAD/CAE 연구실
9/100 - 10/100	2	시스템공학연구소 컴퓨터비젼 연구실
11/100 - 13/100	3	과학기술처
14/100 - 100/100	87	기타 배포기관

제 출 문

과학기술처 장관 귀하

본 보고서를 "Visualization 소프트웨어 개발에 관한 연구(I)"

과제의 연차보고서로 제출합니다.

1992년 7월 20일

주관 연구 기관 : 시스템공학연구소

총괄연구책임자 : 김 문 현

선 임 연 구 원 : 김 동 현

민 병 우

김 경 욱

연 구 원 : 김 기 호

심 숙 희

김 상 원

배 창 석

최 경 호

조 성 익

정 인 숙

요 약 문

Ⅰ. 제목

Visualization 소프트웨어 개발에 관한 연구(I)

Ⅱ. 연구개발의 목적 및 중요성

컴퓨터에서 처리된 정보를 출력하는 기술은 하드웨어의 급속한 발전에 힘입어 단순한 숫자의 출력에서 탈피하여 사람이 자연현상을 보는것과 같이 느끼도록 시각화하여 표현하는 기술로 발전되고 있다. 따라서, 선진외국에서는 Scientific Visualization, 의료,교육, TV등의 영상매체,우주산업, 군사, 예술분야등에서 다양하게 응용되고 있다.이러한 시각화기술은 모델링 및 애니메이션 응용에 적용되어 사용자의 다양한 요구에 부응하는 시각적인 인터페이스 모델을 제공하기 때문에 매우 중요한 요소기술로서, 우리나라에서도 선진국과 같은 정보의 시각화 기술을 실용화하기 위해 요소 기술의 연구 및 개발이 필요하다.

특히, 국내에서는 『대전 EXPO '93』을 계기로 확산되고 있는 컴퓨터 그래픽 애니메이션 제작에 Wavefront, Alias, Explore 등의 선진 외국의 유명 소프트웨어를 고가로 구입하여 사용하는 실정이다. 이와같은 외국산 S/W의 국산화를 위한 연구가 필요하다.

Ⅲ. 연구개발의 내용 및 범위

본 연구는 1991년 8월부터 3개년도를 목표로 하여 수행되는 첨단 요소 분야 연구 과제인 “Visualization 소프트웨어 개발에 관한 연구”의 1차년도의 연구로 기본설계 및 GUI(Graphical User Interface)에 의한 시스템의 일부 모듈을 완성하고, 『3차원 컴퓨터그래픽을 이용한 미륵사지 서탑복원』의 연구과제와 『SERI 건물 애니메이션』의 연구과제에 본 시스템을 사용하여 본 시스템의 문제점을 파악하고, 2차년도의 연구방향을 설정하였다.

Ⅳ. 연구결과 및 적용에 관한 건의

본 연구과제의 결과물이 상업적으로 성공한 외국의 Visualization 소프트웨어보다 모든 면에서 우위를 갖도록하기에는 현실적으로 한계가 있다.

이와같은 이유에서 본 Visualization 시스템은 사용자가 편리하게 사용할 수 있는 환경을 제공하는것과 렌더링 속도 향상등의 두가지 측면에서 외국의 유명 소프트웨어와 비교 우위를 갖는 소프트웨어를 개발한다는 목표를 갖고 본 연구과제를 수행하였다.

렌더링 성능 향상을 위하여 외국 유명 소프트웨어를 분석한 결과 기

존의 외국의 소프트웨어의 경우는 하드웨어 간의 호환성을 고려하여 그래픽 워크스테이션이 갖고 있는 그래픽 엔진의 성능을 활용하지 않고 대부분의 중요한 렌더링을 CPU의 성능만으로 해결하고 있다.

따라서 본 연구에서는 처음부터 기존의 외국 소프트웨어가 가지고 있는 범용성에서 탈피하여 우선적으로 건축분야에의 활용에 촛점을 맞추어 시스템 설계를 하였고, 그래픽 워크스테이션의 그래픽 엔진의 성능을 최대한 발휘할 수 있는 프로그램을 작성하였다. 그 결과 기존 소프트웨어 렌더링 성능의 10배 이상의 속도로 렌더링할 수 있는 시스템을 구축할 수 있었다.

최근의 컴퓨터 그래픽 분야의 기술발전 주기는 6개월 이하로 빨라지고 있다. 컴퓨터 하드웨어의 발달 또한 앞으로 2-3년내에 1000MIPS의 CPU가 개발되어질 것으로 예상되어지고 있다. 이러한 환경의 변화에 대응하기 위해서는 컴퓨터 그래픽 분야의 표준화 경향과 하드웨어 간의 호환성을 고려한 시스템 개발을 하지 않으면 안된다.

본 연구에서 사용한 그래픽 라이브러리인 starbase는 비록 그래픽 표준은 아니지만 현재의 그래픽 표준인 PHIGS 개발의 기본이 된 CGM (Computer Graphics Metafile)에 준거하여 작성된 라이브러리이며 HP사

에서도 Starbase 와 PHIGS를 통합시킨 새로운 라이브러리를 개발하여 곧 공개할 예정이다. 따라서 새로운 그래픽 라이브러리가 등장하더라도 기본 구조의 변경없이 시스템을 재구성할 수 있을 것이다. 또한 본 연구에서 활용한 Graphics User Interface Tool인 OSF/Motif는 워크스테이션 분야에서의 X-window프로그램 개발을 위한 표준 tool로 정착 되어 가고 있고 PHIGS와 X-window를 병합시킨 PEX (PHIGS Extention to X-window)가 새로운 표준화 안으로 결정되어 각사에서 implementation 작업을 진행중이다.

따라서 본 연구에서 사용된 그래픽 라이브러리나 Graphics User Interface는 차세대 표준화 안을 최대한 수용할 수 있는 것으로서 앞으로의 시스템 환경 변화에 충분히 적응할 수 있을 것으로 사료된다.

SUMMARY

I. Title of the Study

"A Study on the Development of Visualization Software(I)"

II. Objectives and Importance of the Study

Until recently interactive computer graphics was not readily applicable because of the expensive display hardware, computer resources, and idiosyncratic software. In the last few years, however, significant reduction in the hardware price/performance ratio, the development of high-level device-independent graphics software packages make pictorial communication natural and efficient. Now in developed countries, computer graphics is being applied in variety of different fields such as scientific visualization, medical imaging, education, image processing, space science, military affairs, and art.

The objectives of this research is to develop and accumulate an essential visualization technique for the interactive modeling and animation. Rencently expensive software such as Wavefront, Alias, Explore have been purchased to prepare computer graphics

animation for 『Taejon EXPO'93』. In this context the development of an efficient visualization softwares is indispensible at this point.

III. Contents and Scope of the Study

This is the first year of 3 year research. This year basic system and graphical user interface (GUI) design of the visualization system have been completed. The contents and scope of this study are 1) concept of visualization system, 2) design and implementation of visualization software, 3) practical application of the visualization system 4) Explanation of the graphical user interface.

1. Concept of visualization system

   General concepts for visualization system were described:

   Concepts of planning, modeling, rendering, preview, recording, editing and dubbing.

2. Development of visualization software

   Basic modules were designed and implemented

3. Practical application of visualization system

   o Graphical restoration of the Pagoda-Miruk Temple.

o Animation of SERI building.

Appendix 1. Explanation of graphical user interface

General overwiew of GUI concept, Xwindow system and OSF/Motif toolkit were described.

Developed visualization software has been applied to the graphical restoration of the Pagoda-Miruk Temple and SERI building. Problems found in this process helped the researchers in setting the direction of the next research.

IV. Results of the Study and Recommendations

It will be very difficult that developing a visualization software which is superior to the commercially available softwares such as Wavefront, Alias, Explore. For this reason, this year's research has been focused on improving rendering speed and implementing user interface which can be superior to other popular visualization softwares. In order to maintain software compatibility some commercial softwares use CPU only without utilizing graphics

hardware engines. However the visualization system we developed has utilized full capability of HP's graphics engine and has been focused on architetural application. The comparision of our visualization system with one commercial graphics software shows that our visualization sytem is 10 times faster than the other one.

To keep abreast with rapid development of hardware technology, visualization system must have capability of compatibility and transferability to the newly developed computers. A CGM (Computer Graphics Metafile) based graphics library named STARBASE was used in this project. Although STARBASE is not a standard graphics library, it seems to be widely used by HP users. HP company is going to announce a new version of graphics libray which unites PHIGS and STARBASE. The visualization system developed in this study can be easily modified to accept this new graphics library without changing the basic structure. For user interface tool OSF/Motif was selected. OSF/motif is industry standard and PEX (PHIGS Extension to X) seems to be a new graphics standard. So graphics library and user interface used in this project will be modified easily to a new system based on X-Window based new standard graphics library.

CONTENTS

목 차

그림 목 차

표 목 차

제1장 서 론

제 1 절 연구의 배경 및 필요성

최근의 컴퓨터그래픽이라는 학문이 각 분야에서 유용하게 적용되고 여러분야에서 각광을 받고있다. 컴퓨터그래픽의 이용분야는 컴퓨터 응용 디자인(Computer Aided Design), 모형비행 연습장치(Flight Simulator), 과학기술 계산의 가시화(Scientific Visualization), 상업광고(Advertizing), 컴퓨터 게임(Computer Game), 컴퓨터 아트(Computer Art)등 다양한 분야에서 연구가 계속되고 있다.

한편, 1993년 대전에서 개최되는 엑스포'93을 계기로 컴퓨터그래픽 기술의 사회적인 관심도가 날로 증가되고 있다. 엑스포'93의 각 기업관에서는 자기회사의 기업홍보를 위하여 수백원의 예산을 들여 전시관을 건설하고 있는 중이며, 그 중에서도 전시관의 꽃이라고 할 수 있는 영상 부문만도 수십억의 예산으로 주로 컴퓨터그래픽 기술을 이용한 영상제작이 기획중 이다. 그러나 이러한 영상등의 제작은 거의 대부분 외국의 컴퓨터그래픽 전문회사에 의뢰하여 제작하는 실정이다. 일부기업에서 국내 전문회사에 의뢰하여 제작하고 있지만 사용하는 시스템이 모두 외국산 하드웨어와 소프트웨어이다.

컴퓨터그래픽을 이용한 영상제작의 해외기술 의존도가 지극히 높은

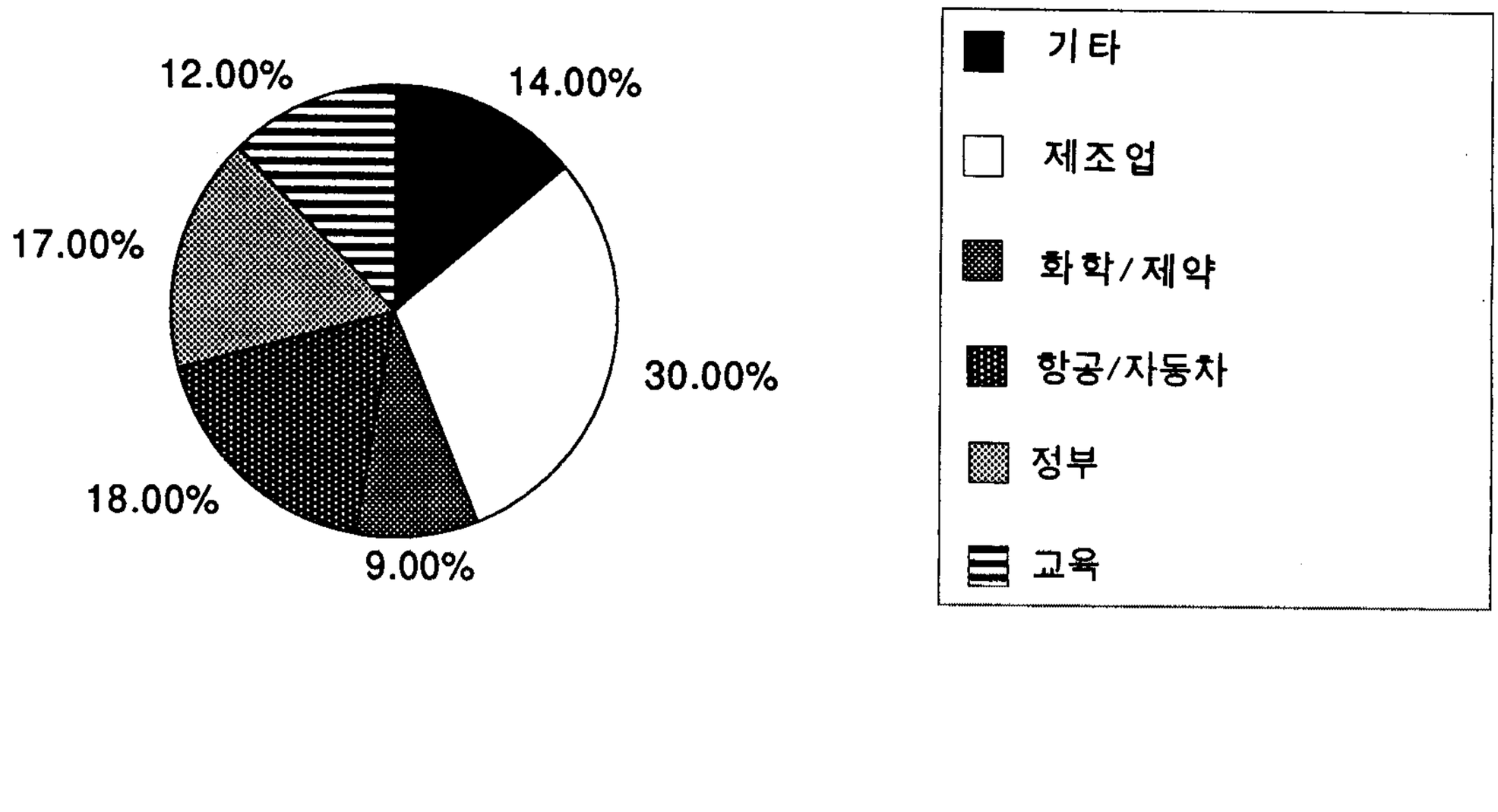
이유는 컴퓨터그래픽이라는 학문이 첨단기술이어서 국내에서는 기초연구가 시작된지 얼마 안되어 국산 소프트웨어의 개발이 전무한 상태이기 때문에 컴퓨터그래픽 응용분야의 모든 핵심기술을 외국에 의존할 수 밖에 없는 것이다.

외국의 S/W를 사용하는 업체들의 공통적인 불만은 국내의 판매처에서는 기술을 전혀 보유하지 않기 때문에 구입직후 실시하는 사용자 기본교육부터 모든 교육을 외국까지 출장을 가서 습득해 와야한다는 사실과 작업시 발생하는 모든 기술적인 사항을 외국의 본사에 문의하여야 하기 때문에 긴급히 처리해야할 사항에 대한 대처가 곤란하다는 것이다. 이와 같은 문제들로 인하여 국내기술에 의한 국산 S/W가 발표되기를 손꼽아 기다리고 있는 실정이다.

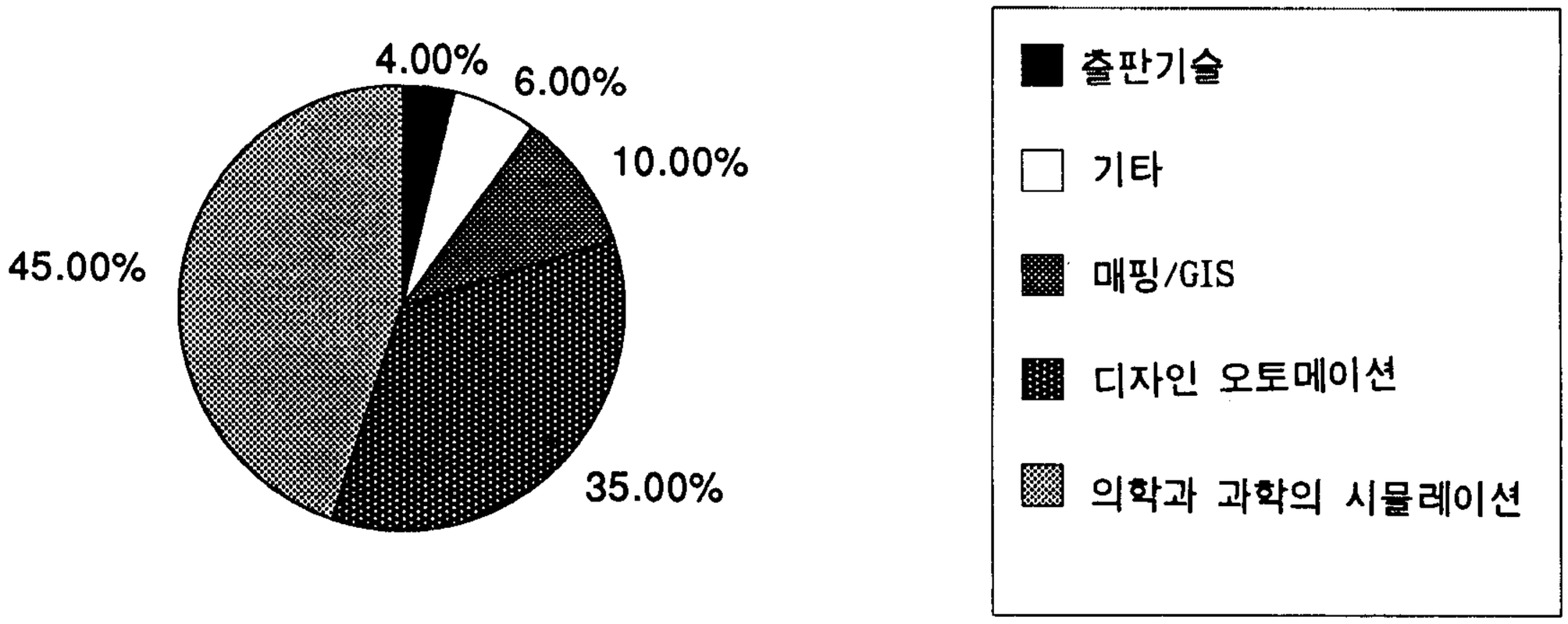
미국의 TFS(Technology Financial Services)의 최근 조사에 따르면 Visualization 워크스테이션에 내장되어 판매되는 Visualization 소프트웨어의 응용분야별, 산업별 Market 분포도([그림 1-1]) 를 보더라도 Visualization S/W의 응용분야중 가장 높은 분야가 의학과 과학 시뮬레이션 분야로 45% 이고, 디자인 오토메이션 분야가 35% 이다. 1차로 국내 대형 건설 프로젝트의 프리젠테이션 영상제작을 위하여 국내의 많은 영상제작 업체에서 필요로 하고 있는 건축분야의 Visualization S/W를 개발하고 있지만 본 연구가 완료되는대로 의학과 과학 시뮬레이션 분야와 디자인

오토메이션 분야의 연구를 실시할 것이다.

산업계 :



응용분야 :



[그림 1-1] Visualization 소프트웨어 Market 분포도

제 2 절 보고서의 구성

제 1 장의 서론에서는 연구의 배경과 필요성에 대하여 논하고, 본 연구 보고서 구성에 대하여 설명하였다.

제 2 장에서는 Visualization 시스템의 최종목표인 3차원 컴퓨터그래픽영상제작을 위한 6단계 - Planning, Modeling, Rendering, Preview, Recording, Editing & Dubbing - 를 위한 각각의 단계에 대하여 설명을 하고 향후에 연구개발되어야 할 High Level Rendering 모듈, Preview 모듈, Recording 모듈에 대한 기본설계를 설명하였다.

제 3 장에서는 1차년도에 개발된 시스템에 대하여 설명을 하였다. OSF/Motif를 사용하여 구현한 GUI(Graphical User Interface)와 시스템의 각각의 모듈별 기능및 메뉴화면을 중심으로 설명하였다.

제 4 장은 1차년도 연구의 결과물을 Test하기 위하여 실시한 시스템 공학 연구소의 3차원 표현과 컴퓨터그래픽기술을 이용한 미륵사지 서탑복원에 대하여 설명하였다.

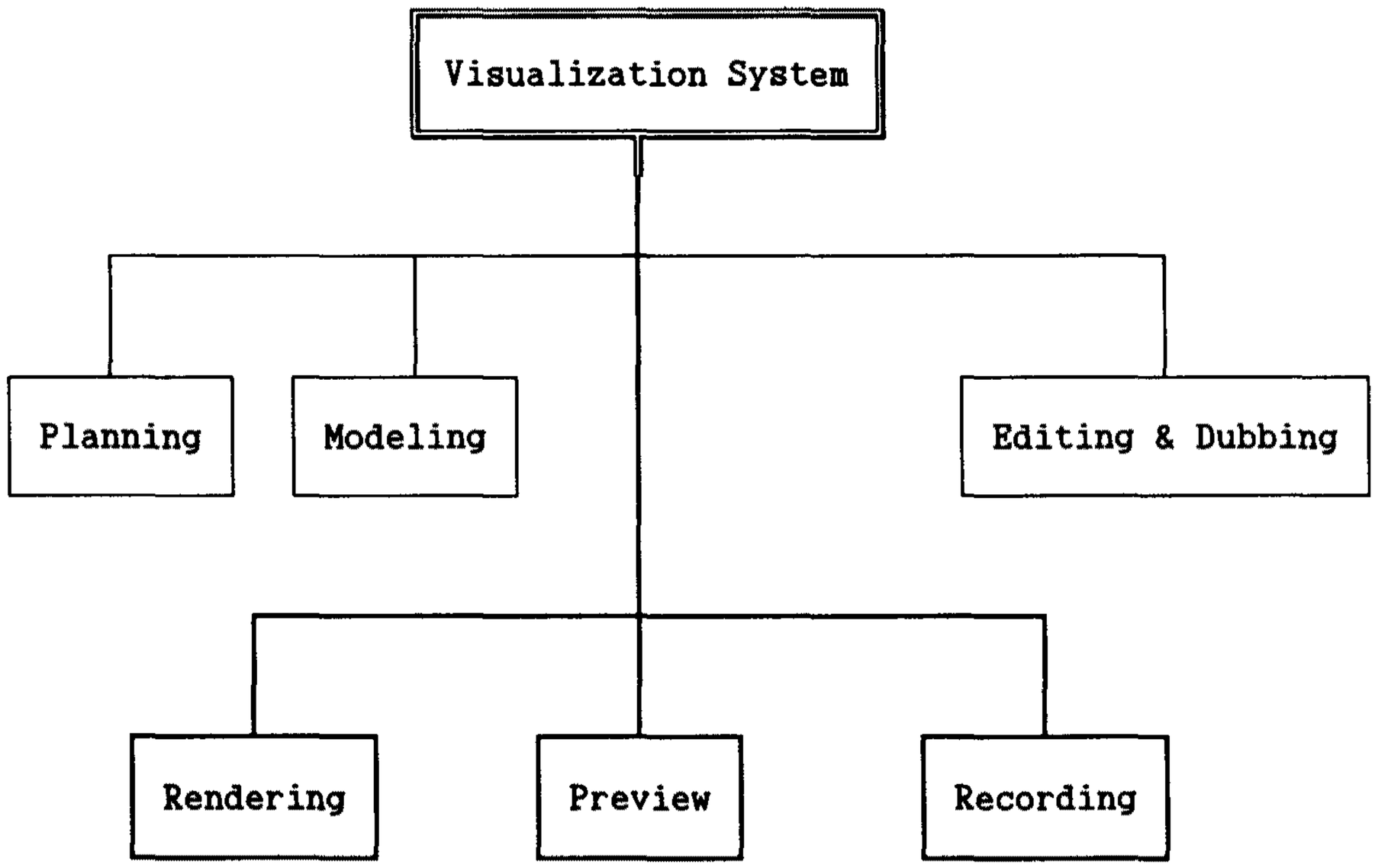
제 5 장 결론 부분에서는 본 연구에 결과의 의의와 개선방향과 앞으로의 전망에 대하여 논하였다.

마지막으로 부록으로 『Graphical User Interface』 대하여 설명을 함으로써 본 연구에서 채택한 X-Window 시스템, X Toolkit, OSF/Motif를 이해하는데 도움이 되도록 하였다.

제 2 장 Visualization 시스템

제 1 절 개 요

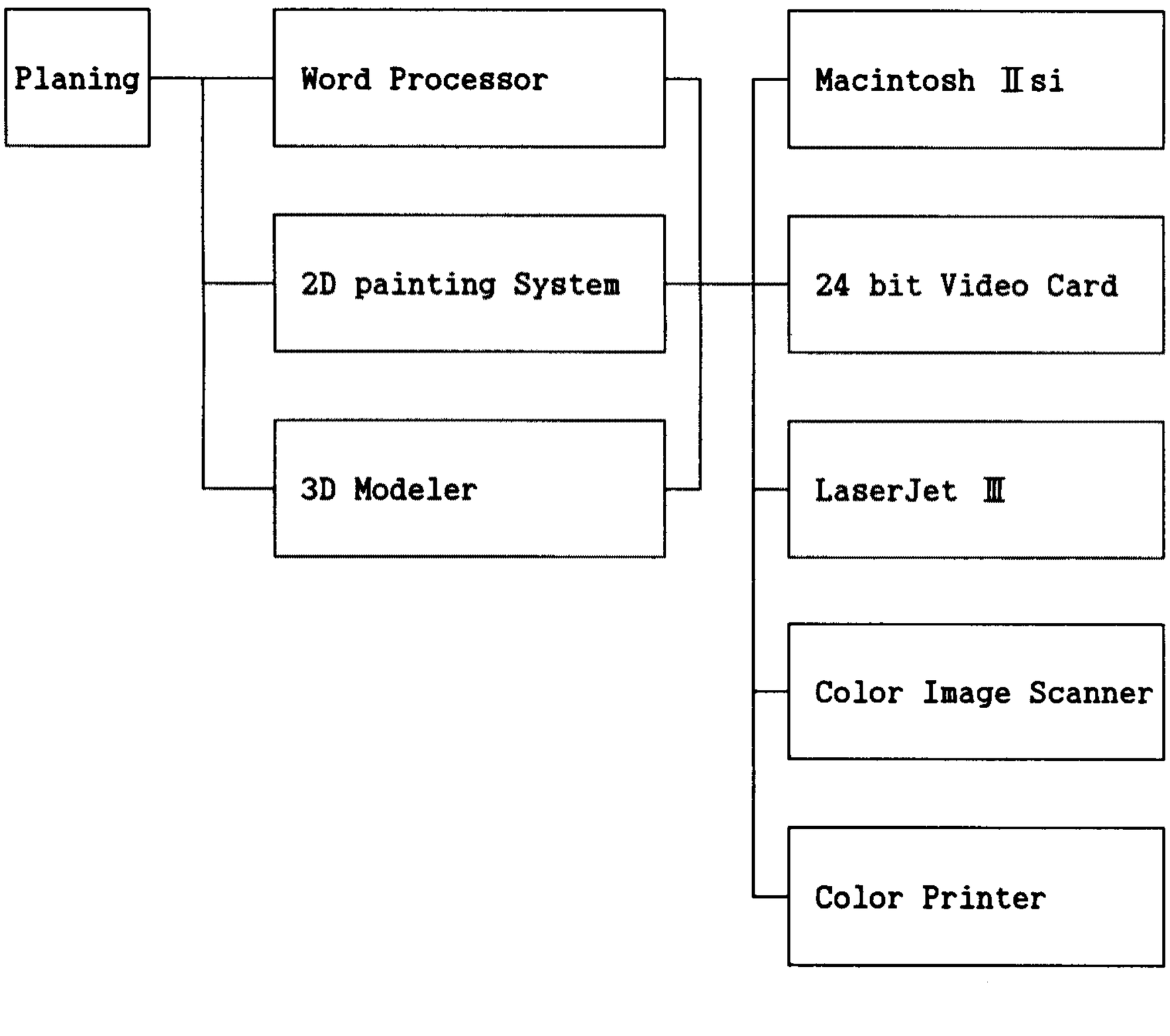
Visualization 시스템은 [그림 2-1]과 같이 Planning, Modeling, Rendering, Preview, Recording, Editing & Dubbing의 6개 모듈로 구성된다. 1차년도 연구는 컴퓨터그래픽 핵심기술인 Rendering 모듈을 중점적으로 연구하고, Preview, Recording 모듈은 2차년도에 연구 수행한다. 그리고, 나머지 모듈은 현재 시판중인 S/W와 H/W들을 연결하여 사용하고, 불편이 발생하는 부분은 자동연결을 위한 프로그램을 작성하여 보완한다.



[그림 2-1] 시스템 구성

제 2 절 Planing 모듈

Planning이란 애니메이션을 제작하기 위한 시나리오의 작성, 스케치(sketch) 시스템을 이용한 콘티(conti) 작성등을 말한다. 또한 시나리오 작성이나 콘티작성은 수주활동에 필수적인 작업으로 프리젠테이션(presentation)의 질이 상당히 중시되는 작업들이다.



[그림 2-2] Planning 모듈의 시스템 구성

이 작업들은 컴퓨팅 파우워(computing power)보다는 고도의 사용자 인터페이스를 요하는 작업으로 도규맨트 작성 중에 콘티 작성으로 옮아 가는등 multi-window, multi-task 기능을 필요로 한다.

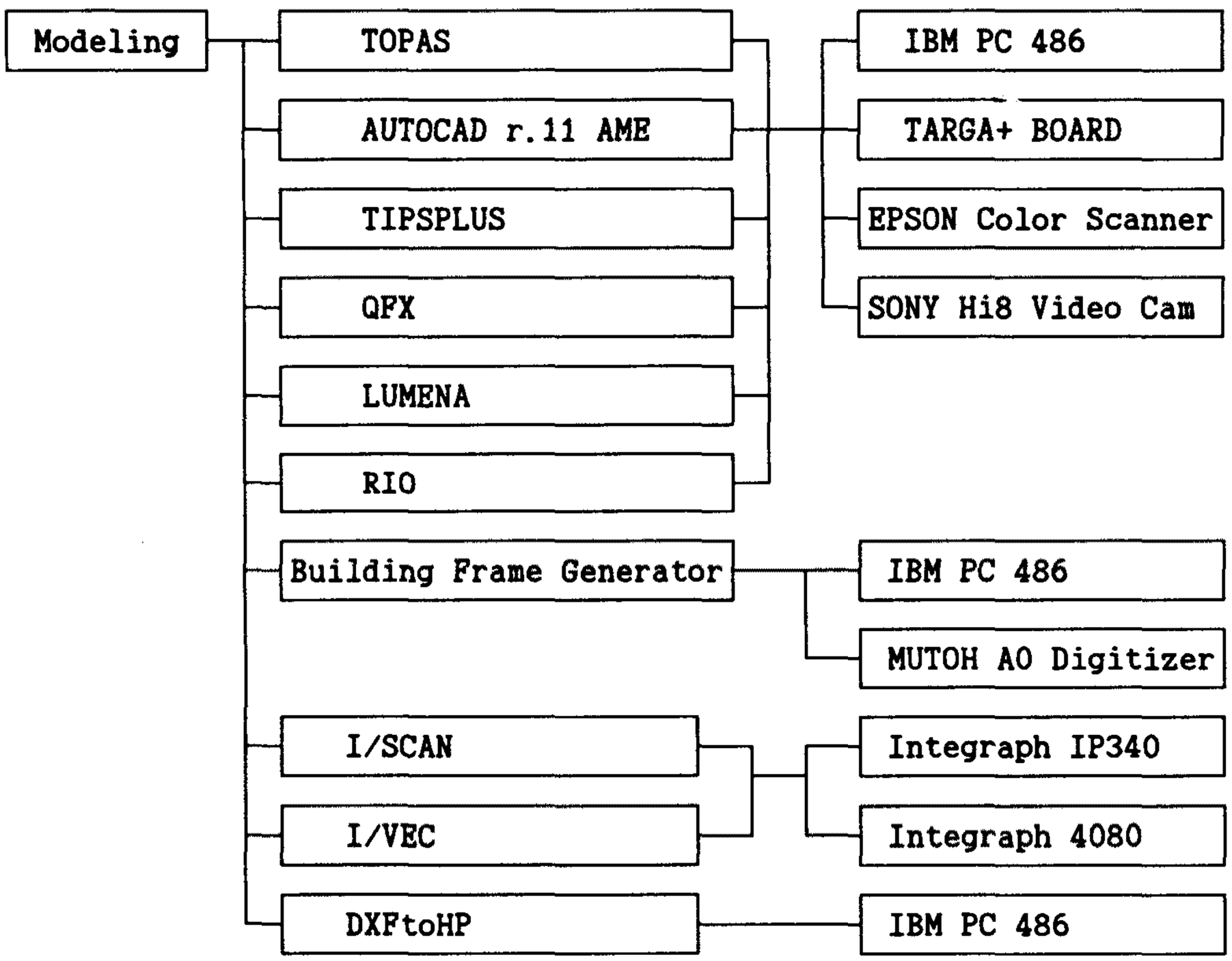
시나리오의 도큐맨테이션은 도형 처리 기능이 포함되어있는 시판 워드 프로세서(word processor)를 사용하는 것이 바람직하다. 콘티작성 작업에는 경우에 따라 기존 일러스트레이션(Illustration)을 스캐닝(scanning)해서 가공하는 방법도 이용되어질 수 있으며, 칼라(color) 표현을 필요로 하는 경우도 있기 때문에 Planning 부문에서는 하드웨어로 full color 대응 매킨토시(Macintosh)를 사용하는 것이 바람직하고, A4 size full 칼라 이미지 스캐너, 칼라 프린터가 필요하다.

소프트웨어로는 매킨토시에서 작동하는 한글 워드프로세서, 2-D 페인팅 시스템(painting system), 3-D 모델링시스템(modeling system) 등을 분석하여 필요한 소프트웨어를 선택하여 본 시스템에 자동 연결되도록 한다.

Planning 작업의 결과는 최종적으로 누구에게인가 프리젠테이션되어야 하기 때문에 마지막 프로세스는 Editing & Dubbing 작업과 유사한 점이 많다. 따라서 Editing & Dubbing 항목에서 좀더 구체적인 설명을 하기로 한다.

제 3 절 Modeling 모듈

모델링(Modeling) 작업은 과학 기술 계산 결과나, 공학 설계 대상물의 가시화를 위한 기초적인 형상 데이터를 작성하는 작업으로, 가장 많은 시간과 인력을 필요로 하는 작업이다. 따라서 고가의 그래픽 워크스테이션을 사용하기보다는 저가격의 퍼스널 컴퓨터를 여러대 구비하여 팀 구성원 각자가 분담하여 형상 데이터를 작성하는 것이 효과적이다.



[그림 2-3] Modeling 모듈의 시스템 구성

하드웨어는 국내에서 가장 보편적으로 사용되어지고 있는 IBM PC를 사용하며, 소프트웨어는 TOPAS, AUTOCAD version 11 AME등의 시판 솔리드 모델러(solid modeler) 소프트웨어를 사용하여 기본적인 형상 데이터를 작성하고, 필요에 따라서는 PATRAN, NASTRAN등의 CAE (Computer Aided Engineering) 소프트웨어로부터 형상 데이터만을 받아서 처리할 수 있는 data exchanger를 작성한다. Texture mapping을 위한 소재의 작성은 퍼스널 컴퓨터에 1,600만색의 표현이 가능한 비디오 카드를 장착하여 2차원 페인팅 소프트웨어를 이용한다. 본 연구에서는 잠정적으로 비디오 카드로는 TARGA+ board를 채택하고, 2차원 페인팅 소프트웨어로는 TIPSPLUS, QFX, LUMENA, RIO등의 성능을 분석하여 적당한 시스템을 선택하기로 한다. 한편, 이 TARGA+ board와 TIPS는 이미지 스캐너와 VTR의 연결이 가능하여 사진등의 소재 뿐만 아니라, 비디오 소재까지도 스캐닝을 하여 가공할 수 있다. 이미지스캐너로는 Hewlett PacKard사의 A4 size Color Image Scanner인 Scan Jet-IIc 를 채용하고, 비디오 스캐너로는 SONY사의 Hi8 video camera V700을 채용한다. 도로, 시가지 데이터등의 작성을 위한 디지타이저는 Mutoh사의 A0 size digitizer를 채용한다. 지형 데이터 입력은 Intergraph사의 Eagle 4080 scanner와 I/SCAN을 이용하여 등고선을 스캐닝하여 I/VEC을 이용하여 vectorizing을 하고, I/RAS를 이용하여 데이터를 수정한 후, vector data를 mesh data로 변환하여 렌더링을 한다.

1. Solid modeler

솔리드 모델러는 이미 각 분야에서 이용되어지고 있는 시판 PC용 솔리드 모델러들의 기능을 분석하여, 적합한 솔리드 모델러를 선택한다. 또한 PATRAN, NASTRAN, MOVIE.BYU 등의 워크스테이션의 CAE 소프트웨어의 솔리드 모델러의 형상 데이터도 처리할 수 있도록 데이터 교환 프로그램을 작성한다.

2. 2D painting system

2D 페인팅 시스템도 역시 시판 PC용 2D 페인팅 시스템들의 기능을 분석하여, 적합한 2D 페인팅 시스템을 선택한다. 현재 사용을 검토중인 시스템은 TIPSPLUS, QFX, RIO, LUMENA등이 있다.

3. BFG (Building Frame Generator)

BFG란 건축, 도시 경관 시뮬레이션에 제한되어 이용되어지는 건물간이 프레임 데이터 작성용 모듈이다. 디지타이저 위에 지적도나 지도를 펴놓고, 건물의 외곽을 입력한 후 건물의 층수, 층고, 지붕의 형태등을 순차적으로 입력하면 자동적으로 건물 간이 프레임 데이터가 작성되어 진다. 이 모듈의 작성을 위해서는 현재 한국에서 건축되어지고 있는 주택들의 표준 지붕 타이프들을 정리하여 기본 데이터베이스를 구축했다.

4. Topology

Topology 모듈은 지형도의 등고선을 스캐너로 입력하여 raster 데이터를 vectorizing한 후, mesh 데이터로 변환시켜서 최종적으로 삼각형 patch로 구성한다. 지형도의 등고선 간격보다 더욱 자세한 표현이 필요할 경우에는 fractal이론을 이용하여 삼각형 patch 내부의 자세한 지형의 근사치를 추출하여 표현했다.

5. Data exchange

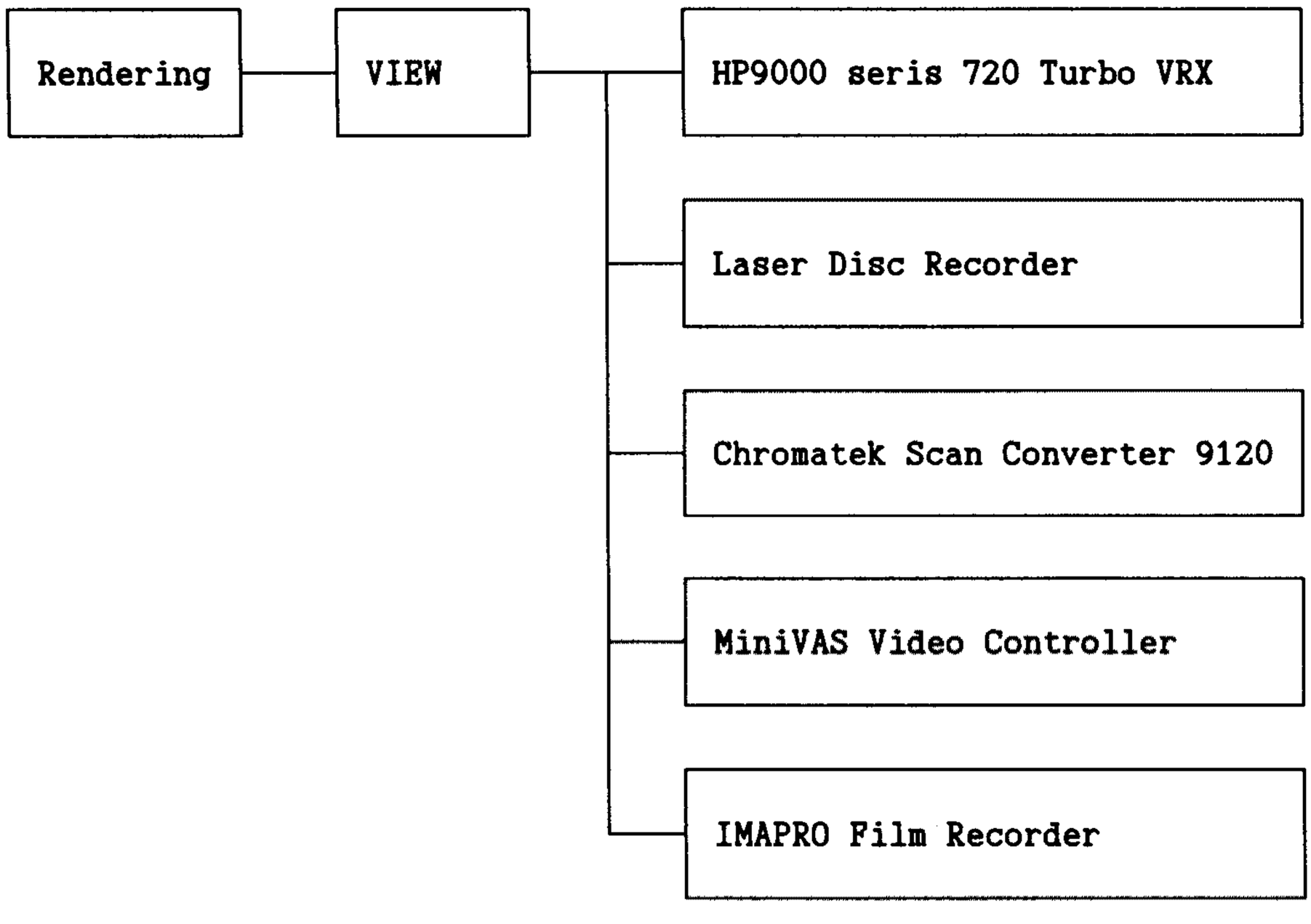
Data exchange 모듈은 TOPAS, AUTOCAD, VERSACAD, ME30, ABACUS, PATRAN, NASTRAN, INTERGRAPH, GDS등 다양한 모델러의 데이터를 활용하기 위한 데이터 교환 필터 기능을 한다.

현재 CAD (Computer Aided Design) 분야의 데이터 교환 형식으로 많이 사용되어지고 있는 것으로는 DXF (Data eXchange File)와 IGES-3D (Initial Graphics Exchange Specification) 등을 이용할 수 있다.

본 시스템에서 사용하는 형상 데이터 구조는 컴퓨터 그래픽 분야에서 사용하는 가장 기본적인 데이터를 사용하고 있기 때문에 커다란 지장없이 각 modeler로부터 데이터 교환이 이루어질 수 있을 것으로 예상되며 사용에 약간의 불편한 부분은 부분적으로 보완했다.

제 4 절 Rendering 모듈

랜더링(Rendering)이란 scientific visualization 시스템의 핵심이 되는 그래픽 표현 모듈로서, 컴퓨팅 파우어(computing power)뿐만 아니라 다양한 그래픽 처리 기능은 물론 interactive한 유저 인터페이스 기능을 보유한 컴퓨터를 필요로 한다. 현재 시판되어지고 있는 컴퓨터중에서 이러한 조건을 가장 만족시켜주는 것이 그래픽 워크스테이션이라고 할 수 있다.



[그림 2-4] Rendering 모듈의 시스템 구성

최근에 발표된 그래픽 워크스테이션에서는, CPU 뿐만 아니라 그래픽 엔진에서도 RISC(Reduced Instruction Set Computer) chip이 쓰여지고 있어서, 그래픽 처리 속도만으로는 수퍼 컴퓨터에 필적하는 성능을 발휘하고 있다. 현재 시판되어지고 있는 그래픽 워크스테이션중에서 그래픽 성능이 가장 뛰어난 기종으로는 Hewlett Packard사의 HP 9000 series 750 Turbo VRX, Silicon Graphics사의 IRIS Crimson, IBM사의 RS6000 등을 들 수 있다.

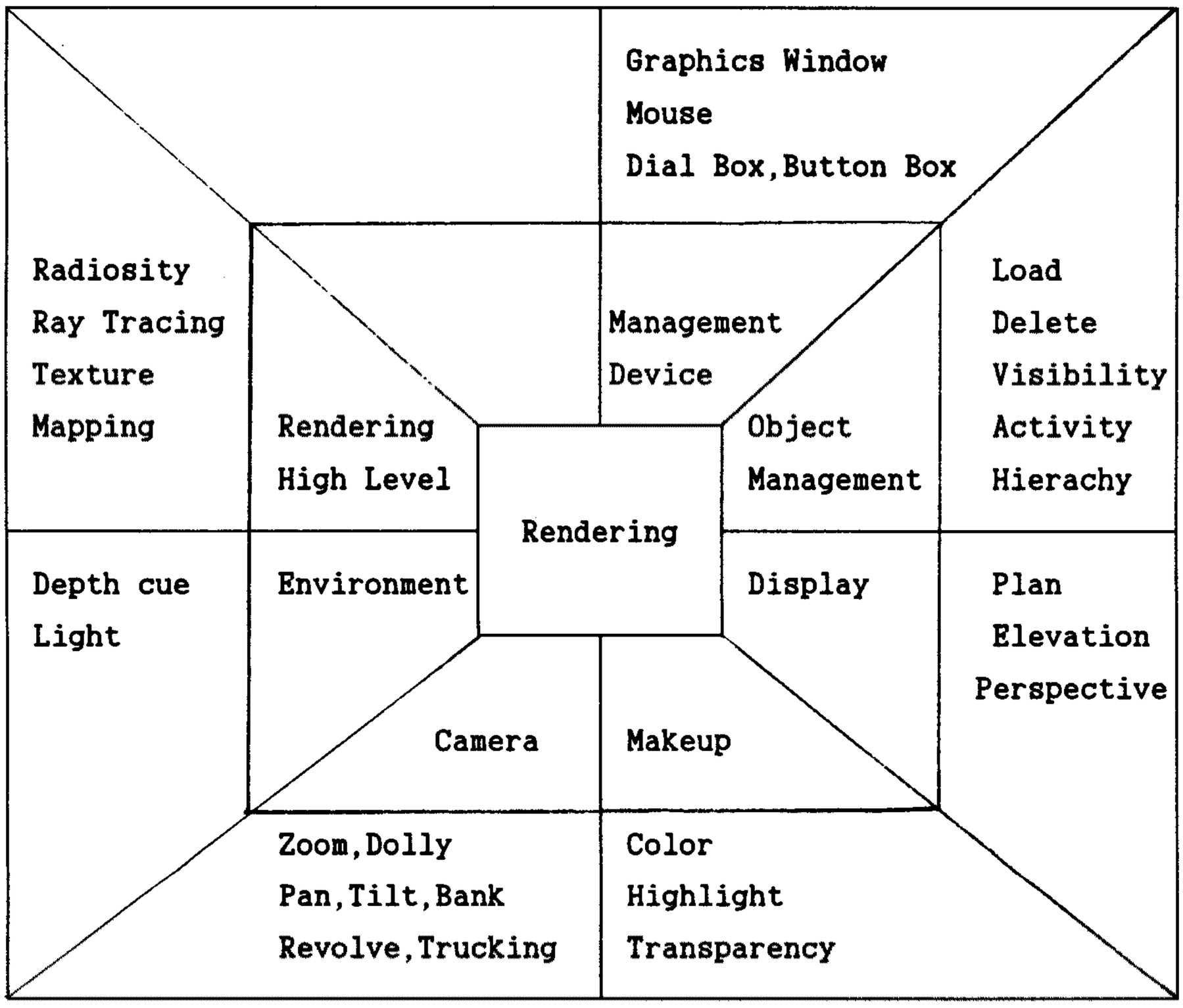
본 연구에서는 HP9000 series 720 Turbo VRX를 채용하여 렌더링 모듈을 구축했다. 이 모델의 그래픽 엔진인 Turbo VRX는 i860을 탑재한 transform 엔진을 2개에서 4개까지 장착하여 병렬로 작동시킬 수 있다.

한편, 하드웨어의 다양화를 꾀하기 위하여 본 시스템의 일부를 IRIS의 low end version인 IRIS Indigo에도 이식하여 장래에 IRIS Crimson에 이식할 수 있도록 한다. RS6000의 경우는 IRIS의 그래픽 엔진을 그대로 쓰고 있기 때문에 일단 IRIS에 이식이 되면 RS6000에 이식하는 것도 어렵지 않게 된다.

1. Device management

Device management 모듈에서는 X-window의 graphics window, mouse, dial box, button box등의 디바이스를 open, close시키는 일을 한다.

각 디바이스에 관한 서브루틴은 렌더링 부분의 각 모듈에서 call하여 사용할 수 있도록 했다.



[그림 2-5] Rendering 모듈 서브루틴

2. Object management

Object management 모듈에서는 geometric data의 load, delete, visibility control, activity control, data hiearchy의 조정등의 일

을 한다. 데이터의 load는 C language의 구조체 및 pointer 기능을 최대한 활용하여 메모리내에 데이터베이스를 구축하여 메모리 절약 및 데이터 베이스 검색의 신속성등을 꾀한다.

Activity control이란 물체에 대한 makeup data의 변경등을 할때에 변경 해당 물체를 선택하는 기능을 일컫는 것으로 데이터의 hierachy를 이용하여 다수의 물체를 동시에 선택할 수도 있도록 한다. Object의 list는 X-window의 메뉴상에 나열하여 마우스에 의한picking 조작만으로 선택이 가능하도록 했다. Data의 hierachy는 자유자재로 그 구조를 변경할 수 있도록 했다.

3. Display

Display 모듈은 본 시스템의 핵심 부분으로 plan, elevation, perspective, axonometric등의 다양한 투영법에 대응한 그래픽 윈도우를 multi-window로 open시켜서, visibility on이 설정되어 있는 물체를 표시했다. Option에 의하여 plan 및 section에 현재의 perspective의 시점 및 참조점의 위치 카메라의 화각등을 표시하도록 했다.

Plan 및 section 상에서의 interactive한 조작을 위해서는 redraw를 신속하게 하지 않으면 안된다. 이를 위해서는 물체의 load시에 각 물체의 x,y,z의 최대, 최소값을 계산하여 두고, rendering이 중시되지 않는 조작에서는 이 값에 의한 3차원 큐브를 생성하여 디스플레이하도

록 하여 디스플레이 시간을 절약하고, rendering이 필요한 경우에만 원래의 데이터를 이용하여 display하도록 했다.

Display 모듈에 필요한 기본 rendering 기능으로는 coloring, shading, highlight, transparency, lighting등이 있다.

한편 필요에 따라서 wireframe이나 surface model로 표현하는 방법을 선택할 수 있도록 했다.

4. Makeup

Makeup 모듈에서는 activity on이 설정되어 있는 물체의 color, highlight, transparency등의 속성 데이터를 변경하는 일을 한다.

Color의 변경은 RGB, HSL의 어느 쪽으로도 변경이 가능하도록 하며, color table을 이용하여 redraw를 하지 않고도 물체의 color를 변경할 수 있는 방법으로 프로그래밍했다.

5. Camera

Camera 모듈에서는 카메라의 위치, 주시 방향, 이동등을 설정한다. 카메라의 위치 설정은 plan과 elevation에서 마우스에 의한 위치 설정, 키보드에 의한 좌표점 입력, camera data의 load 등의 방법 중에서 적당한 방법을 사용자가 선택하여 작동할 수 있도록 했다.

마우스에 의한 입력 방법은 plan에서는 시점 및 참조점의 x,y 좌표를, elevation에서는 시점 및 참조점의 z 좌표를 마우스에 의해 pick

할 수 있도록 했다.

카메라 이동은 카메라 촬영 기법의 기본 동작인 pan, zoom, tilt, revolve, bank, trucking, dolly등을 메뉴에 표시하여, 마우스에 의해 선택하는 방법과, dial box를 직접 조작하여 이동하는 방법중에서 적당한 방법을 사용자가 선택하여 작동할 수 있도록 했다.

Pan이란 카메라의 위치를 고정시키고 카메라의 방향을 좌우로 회전시키는 방법을 말한다.

Zoom이란 카메라의 촛점거리를 변화시키는 방법으로, 촛점거리를 증가 시키는 것을 zoom in, 촛점거리를 감소시키는 것을 zoom out이라고 한다.

Tilt란 카메라의 위치를 고정시키고 카메라의 방향을 상하로 회전시키는 방법으로 tilt up, tilt down이 있다.

Revolve란 피사체를 중심으로 카메라의 위치를 상하좌우로 회전이동시키는 방법이다.

Bank란 카메라와 피사체를 연결하는 축을 중심축으로 카메라를 좌우로 회전시키는 방법이다.

Trucking이란 피사체에 평행하게 카메라를 좌우로 이동하는 것을 말한다.

Dolly란 카메라가 피사체에 접근하거나 멀어지는 것을 말하며, 전자

를 dolly in, 후자를 dolly out이라고 한다.

6. Environment

Environment 모듈은 물체가 놓여져 있는 주변 환경 즉 light, depth cue 등의 데이터를 정의 또는 변경하는 모듈이다.

Light의 설정은 light data를 읽어서 설정하는 방법과 화면상에서 직접 설정하는 방법이 있다. 화면상에서 직접 설정할 경우, 현재의 설정치를 숫자로 표시하여 수정이나 추가가 필요한 데이터를 입력한다.

Light의 위치에 관한 정보는 perspective 스크린상에 도형화하여 표시하여 설정치 확인에 도움을 줄 수 있도록 했다.

일반적으로 그래픽 워크스테이션의 lighting은 한 물체에 대하여 8개에서 16개까지의 light 설정이 가능하고, 각 물체에 대하여 서로 다른 별도의 light set를 설정할 수 있다. 따라서, light 설정 모듈에서는 각 물체별로 light 설정이 가능하도록 하여야 하며, 새로운 light set의 추가 삭제 등도 간단한 조작에 의해 가능하도록 하였다.

Depth cue란 원근감의 표현에 효과적으로 이용되어질 수 있는 기능이다. 즉, 시점에서 가까운 물체는 원래의 색 그대로 보이지만, 시점에서 멀어질수록 점점 뿌영게 보이게 된다. 즉, Z-buffer의 앞쪽에서 뒤로 갈 수록 원래의 색에 일정한 비율로 다른색을 합성시켜서 표시

하면 되는 것이다. 이 depth cue에 필요한 parameter는 depth cue의 시작과 끝나는 범위를 지정하는 hither면 및 yon면의 거리, hither면과 yon면의 color등 이다.

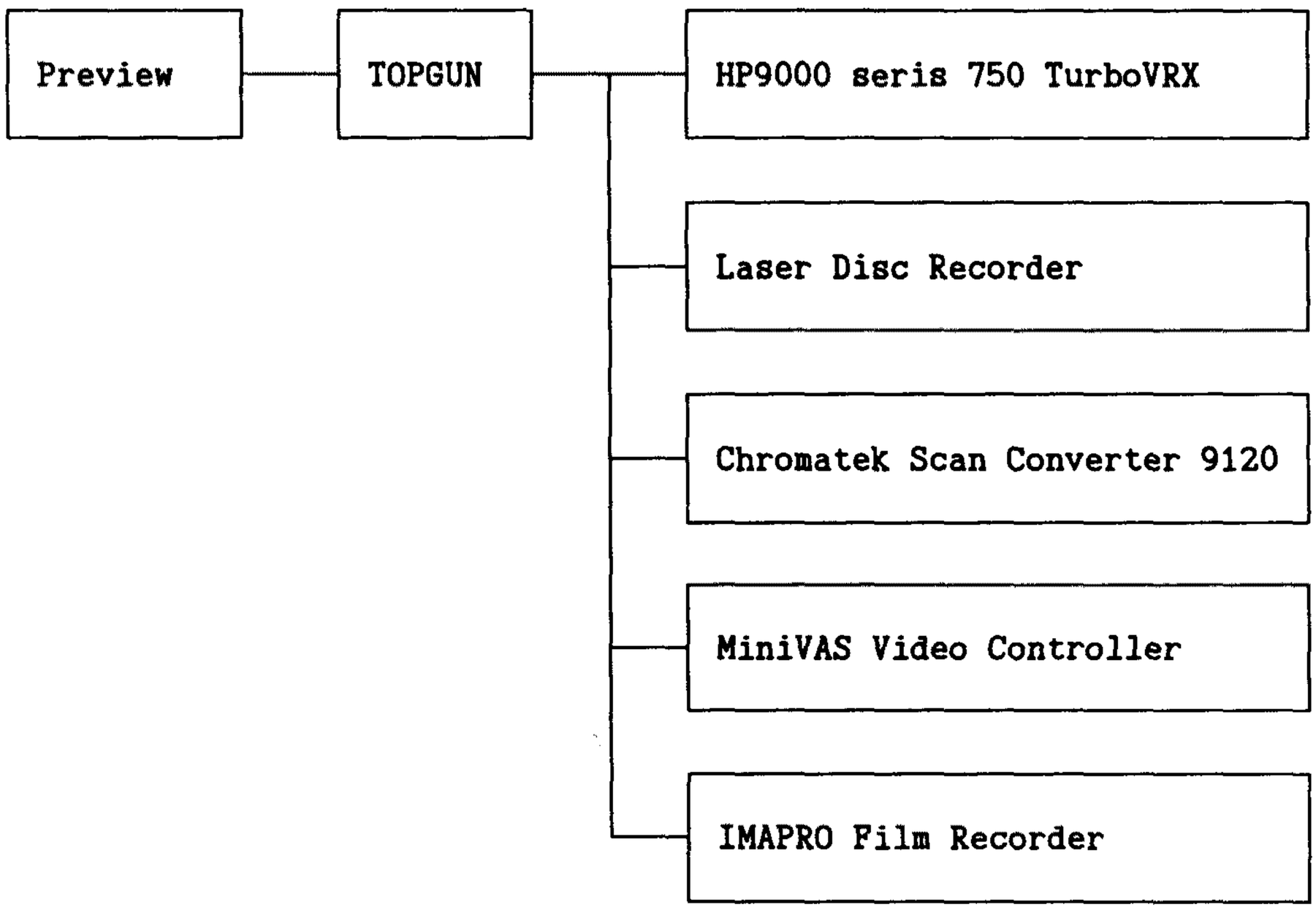
이러한 hither면과 yon면의 설정 역시 plan과 elevation상에 도형화하여 표시하여 마우스에 의해 위치를 지정할 때에 정확한 위치를 파악하는 데 도움되도록 하였다.

7. High level rendering

High level rendering 모듈은 radiosity, ray tracing, texture mapping, mixing 등의 rendering을 수행하는 모듈이다. 이러한 기능들은 상당한 처리 시간을 요구하는 기법들이기 때문에 일단 계산이 끝나서 화면에 표시된 영상은 그 자리에서 비트맵 데이터 형식으로 저장할 수 있도록 프로그래밍하였다. 또한 radiosity의 경우에는 ray tracing이나 texture mapping등과는 달리 display list의 내부 데이터를 변경시켜서 최종적으로 비트맵 데이터를 생성하기 때문에 display list를 저장하는 기능도 필요하게 된다.

제 5 절 Preview 모듈

Preview란 시나리오에 의해 애니메이션을 작성할 수 있도록 key frame의 3차원 공간 내의 시점 및 참조점을 설정하여 시간축에 따라 한 key frame 으로부터 다음 key frame까지 각 시점 및 참조점이 자연스럽게 이동할 수 있도록 보간하여 시점 데이터를 작성하는 시스템이다.



[그림 2-6] Preview 모듈의 시스템 구성

시점 데이터 작성의 다음 단계인 레코딩은 상당한 시간을 필요로 하는 작업이기 때문에, 레코딩에 들어가기 전에 시점의 움직임을 검토하여 이상이 있을 경우 시점 데이터를 재작성하여 움직임에 이상이 없다고 판단되어질 때 레코딩을 시작하지 않으면 안된다.

이러한 시점의 움직임을 검토하는 시스템은 랜더링에서와 같이 다양한 그래픽 기능을 구사할 필요는 없지만 실제 시간으로 각 frame을 묘사하여야 하기 때문에 그래픽 워크스테이션의 3차원 geometric transform 성능을 최대한 활용하여 시스템을 구축하여야 한다.

따라서 preview 역시 랜더링 모듈과 마찬가지로 HP 9000 series 720 Turbo VRX를 이용하여 작성한다. 타 기종에의 이식 문제는 그래픽 기능의 가장 기본이 되는 line 기능만을 사용하기 때문에 커다란 어려움 없이 이식이 가능하다.

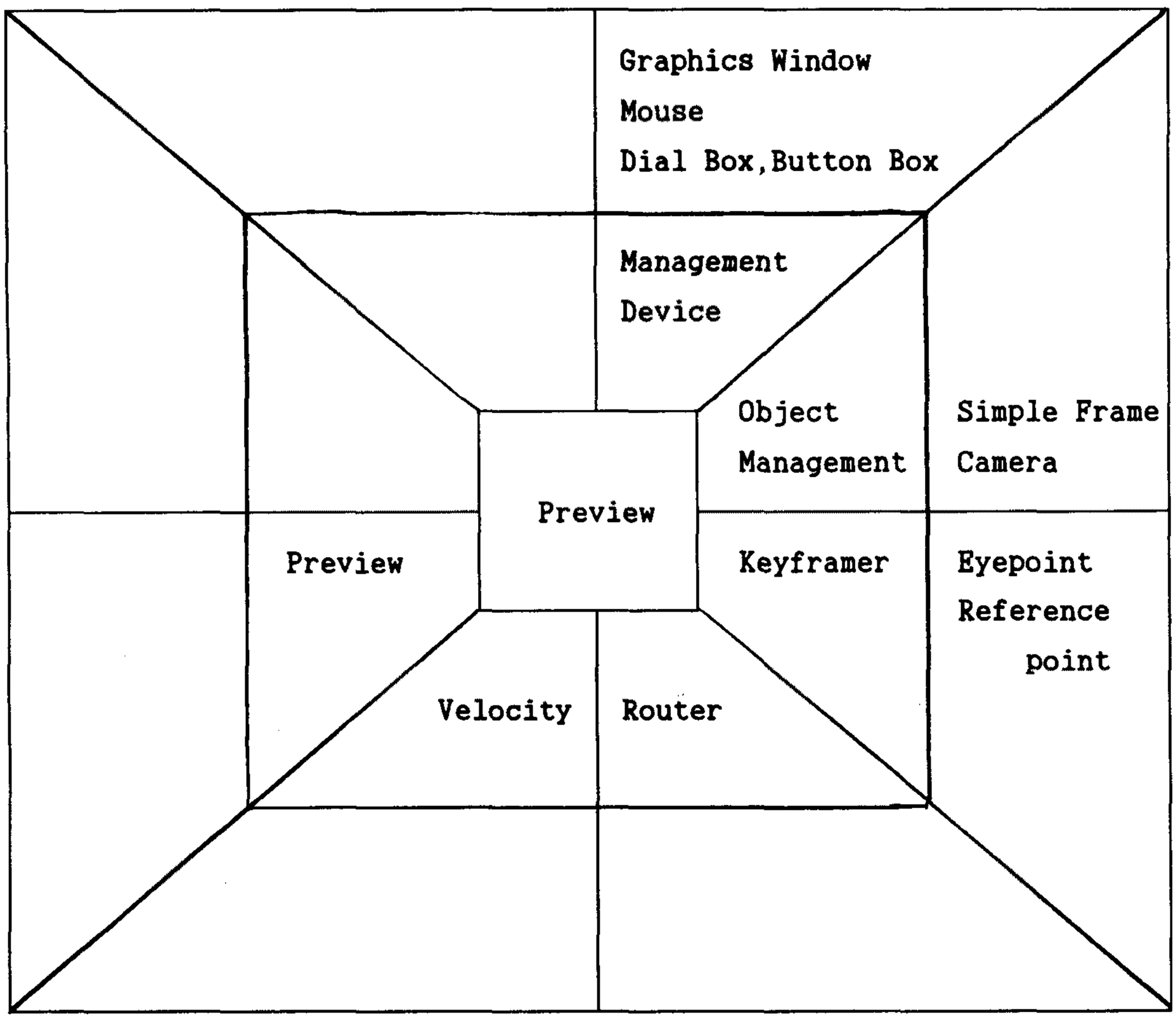
1. Device management

Device management 모듈에서는 X-window의 graphics window,mouse, dial box, button box등의 디바이스를 open, close시키는 일을 한다.

각 디바이스에 관한 서브루틴은 preview 부분의 각 모듈에서 call하여 사용할 수 있도록 한다.

2. Object management

Object management는 preview를 위한 간이 형상데이터의 입력 및 시점 데이터의 출력을 관리하는 모듈이다. 입력 부분에서는 preview만을 위하여 별도 또는 자동으로 작성된 간이 형상 데이터의 입력을 수행한다.



[그림 2-7] Preview 모듈 서브루틴

3. Key framer

Key framer란 CG 애니메이션의 시나리오에 맞는 주요 프레임에 대한 시점 및 참조점 데이터를 추출하는 모듈이다. 적당한 프레임을 잡기 위한 시점 및 참조점의 이동은 랜더링 모듈에서의 시점 및 참조점 이동방법을 그대로 이용할 수 있다.

4. Router

Router란 Key framer에서 결정된 각 Key frame의 시점 및 참조점 데이터 사이를 곡선에 의해 부드럽게 연결시켜 주는 모듈이다.

이용 가능한 곡선은 B-Spline 곡선, N-Spline 곡선, 3차 Spline 곡선, Bezier 곡선등 있다. 일반적으로 이러한 곡선들을 제어하는 control point는 곡선 위를 통과하지 않기 때문에 key frame 위를 통과하고 Key frame에 의해 곡선을 제어할 수 있는 방법을 연구하여야 할 필요가 있다.

5. Velocity

Velocity란 시점 및 참조점의 이동 속도를 설정하는 모듈이다.

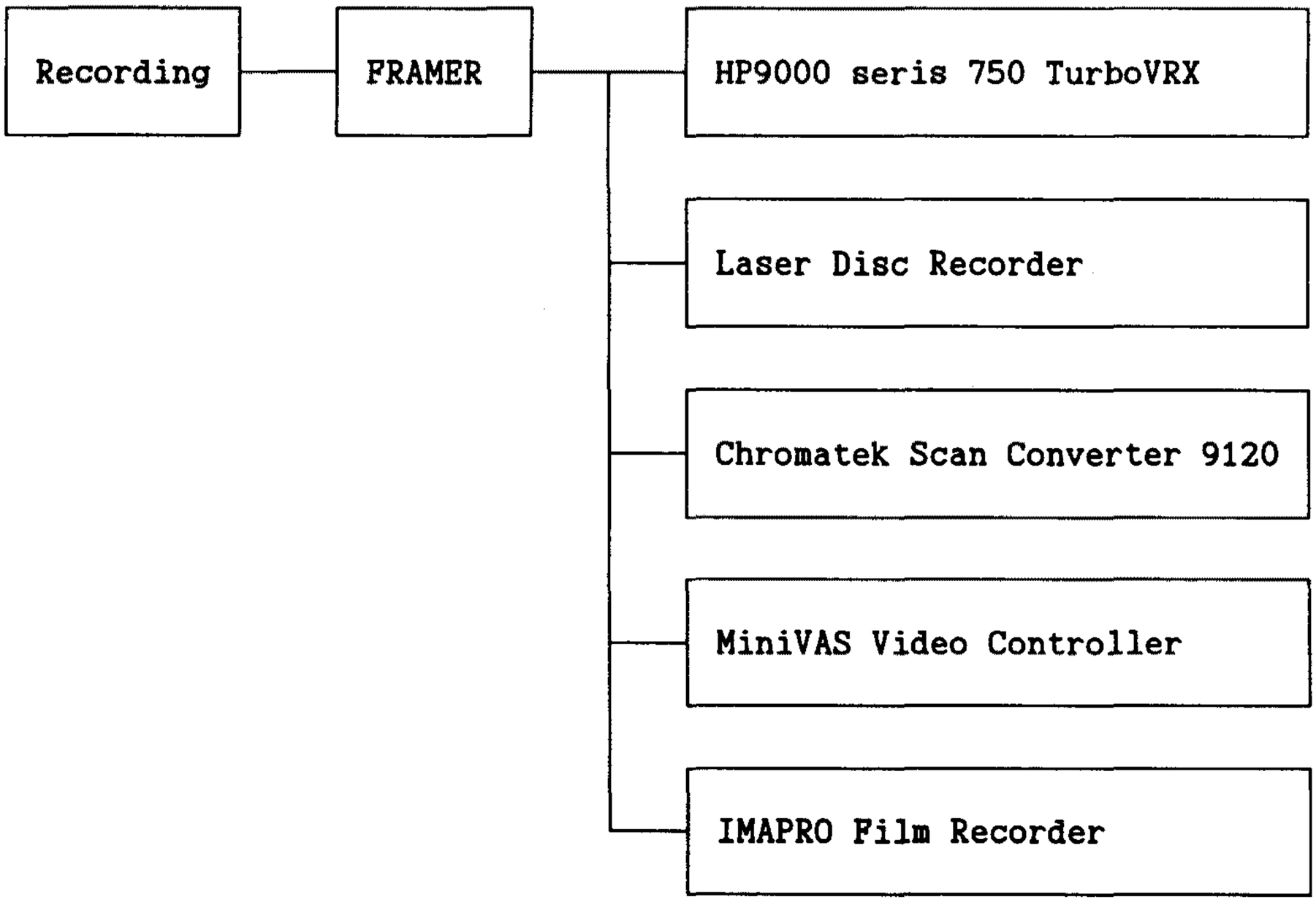
시점 루트상에서 속도 변화가 발생하는 점을 설정하여 속도를 입력하여, 각 점사이를 등가속으로 연결시켜 나가면서 시점 데이터를 작성한다.

6. Preview

Preview란 router와 velocity에 의해 작성된 시점 데이터에 의해 실제의 애니메이션의 움직임을 사전에 검토하는 모듈이다. 간이 형상 데이터를 wireframe으로 double buffer 기능을 이용하여 표시하면 거의 real time으로 애니메이션을 표현할 수 있다. 데이터의 양에 따라 표시 시간이 달라지는 문제를 해결할 수 있는 방법을 모색하여야 한다.

제 6 절 Recording 모듈

레코딩이란 전술한 프로세스를 거쳐 제작된 데이터를 활용하여 비디오 테이프나 영화 필름에 한 프레임씩 녹화해 가는 작업을 말한다. 이 작업은 크게 두가지 방법에 의해 진행되어질 수 있다. 첫번째 방법은 한 프레임씩 렌더링을 해 가면서 녹화하는 방법이고, 다른 방법은 미리 렌더링된 화면의 비트맵 데이터를 다른 그래픽 워크스테이션에 전송하여 순차적으로 녹화하는 방법이다.



[그림 2-8] Recording 모듈의 시스템 구성

첫번째 방법은 그래픽 워크스테이션이 한대 밖에 없을 때 유효한 방법이고, 두번째 방법은 두대 이상의 그래픽 워크스테이션을 이용하여 작업을 분산시키므로서 전체적인 작업 시간의 단축을 꾀할 때 유효한 방법이다. 외견상으로는 두대 이상의 그래픽 워크스테이션을 이용하는 후자의 방법이 당연히 전체 시간을 단축시킬 수 있지만 워크스테이션간에 네트워크를 이용한 데이터 전송 시간이 추가되기 때문에 그래픽 워크스테이션의 절대 사용 시간은 전자의 쪽이 유리하다. 따라서, 어떠한 방법을 채용할지는 해당 조직의 경제적 인적 여건을 고려하여 신중히 선택되어져야 한다.

다음으로 랜더링에 VTR (Video Tape Recorder) 또는 Laser Disc Recorder를 사용할 경우, 그래픽 워크스테이션의 비디오 신호인 RGB 신호를 VTR의 비디오 신호인 NTSC 신호로 변환하지 않으면 안된다.

이때에 양쪽의 화면의 해상도가 틀리기 때문에 화상을 압축하는 작업도 병행되어져야 한다. 이러한 작업을 실제 시간으로 처리하는 것이 scan converter이다.

본 연구에서는 Yamashita의 real time scan converter CVS-960A 또는 Chromatek의 real time down converter 9120의 가격 및 성능을 비교하여 선택하도록 한다.

한편, VTR을 이용하여 한 프레임씩 녹화를 할 경우, VTR을 자동으로

제어하는 video controller가 필요하다. 이 video controller는 컴퓨터의 표준 interface인 RS-232C connector를 통해 보내온 신호를 받아서 VTR의 제어 interface인 RS-422 connector를 통해 VTR에 제어 신호를 보내는 장치이다. 본 연구에서는 MiniVAS의 LYONLAMB을 채용한다.

녹화 방법은 VTR을 이용하는 방법과 영화 필름을 이용하는 방법이 있다. 영화 필름을 사용할 경우는 촬영이 완성되어 현상소에서 현상이 끝날 때까지 촬영이 제대로 되고 있는지 체크할 수 없고, 현상 과정에서 현상액의 농도 및 온도에 따라 필름 세트별로 색이 달라지거나 단순한 실수에 의해 현상에 실패하는 경우도 발생할 수 있다.

따라서, 최근에는 대부분의 CG 프로덕션들이 VTR을 이용하여 녹화하는 방법을 채택하고 있고, 영사기를 이용하여 스크린에 투사할 필요가 있을 때에는 VTR 화면을 필름으로 변환하는 kineco 기법을 이용하여 필름을 제작하며, 특별히 대형 스크린에 투사할 필요가 있을 때에는 고해상도 film recorder를 이용하여 촬영을 하거나, 화면을 직접 촬영하는 방법을 채택하기도 한다.

VTR의 종류로는 VHS, S-VHS, U-matic, betacam, 1 inch VTR, laser disc등이 주로 사용되어지고 있다. VHS 및 S-VHS는 일반 가정용 VTR의 format이기 때문에 방송용으로 사용하기에는 화질이 떨어진다. 최근에는 방송국용으로는 화질이 좋고 들고 다니기 간편한 betacam방식이 널리 이

용되어지고 있다. 그러나, 테이프를 이용한 녹화 방법은 한 프레임을 녹화하기 위하여 반드시 5초 이상의 pre-roll이라는 과정을 거쳐야 하고, 장기간 보존이 불가능하며, 복제를 계속하는 과정에서 화질이 현저하게 떨어진다는 문제점이 있다. 이러한 문제점을 보완하기 위해 개발된 것이 digital 녹화 방식의 laser disc recorder이다. 이제까지 개발된 laser disc recorder는 추기형 녹화 방식으로 한번 녹화된 부분은 수정이 불가능하다. 최근에는 저가격의 rewritable laser disc recorder가 개발되어 이러한 문제가 해결되어 있다.

이러한 점에서 본 연구에서는 rewritable laser disc recorder를 채용한다.

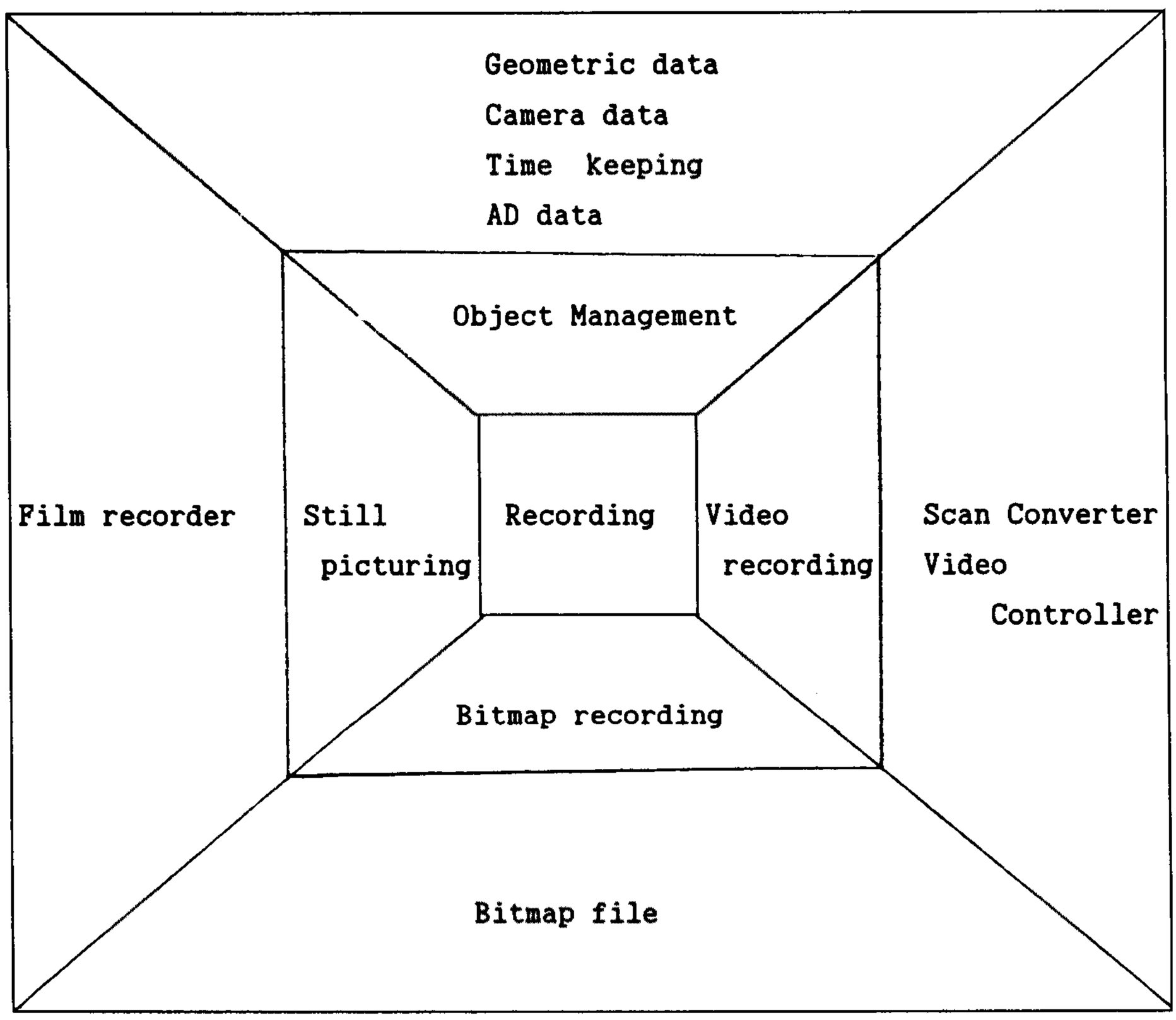
1. File I/O

File I/O 모듈에서는 camera data, time keeping data, AD data등의 load 및 데이터베이스 관리를 한다.

2. Video recording

Video recording 모듈에서는 그래픽 워크스테이션에서 rendering된 RGB 신호 영상을 Video의 영상 신호인 NTSC 신호로 바꾸어 Video Recorder로 보내고, 녹화 개시 신호를 RS-232C interface를 통해 Video Recorder로 보내어 1 frame씩 순차적으로 녹화하는 일을 한다.

RGB 신호를 NTSC 신호로 바꾸어 주는 일은 scan converter가 real time으로 하고, Video Recorder에 녹화 개시 신호 및 녹화 address등을 workstation으로 부터 RS-232C interface를 통해 받아서 RS-422 interface를 통해 Video Recorder로 보내는 일은 video controller가 한다.



[그림 2-9] Recording 모듈 서브루틴

3. Bitmap recording

Bitmap recording 모듈에서는 rendering된 영상의 bitmap data를 file 에 저장하거나, file에 저장되어 있는 bitmap data를 load하여 VTR로 녹화하는 일을 한다.

4. Still picturing

Still picturing이란 rendering된 영상의 bitmap data를 file에 저장 하거나, file에 저장되어 있는 bitmap data를 load 하여 화면을 카메라로 촬영하기 위한 모듈이다.

제 7 절 Editing & Dubbing 모듈

Editing이란 각 sequence별로 녹화된 소재를 시나리오에 맞게 편집하는 것을 말한다. 이때에 영상 효과를 살리기 위해 fade in, fade out이나 chroma key 합성등의 다양한 특수 효과를 삽입할 수도 있다. Dubbing이란 음향 효과를 삽입하는 것을 말한다.

Editing을 위한 장비로는 editing control unit, 영상 mixer인 chroma keyer, 자막이나 그래픽등을 삽입하는 graphic telloper, 화상의 색을 조절할 수 있는 color corrector, 화면의 wipe 패턴을 조작할 수 있는 wipe pattern extender등이 있고, Dubbing을 위한 장비로는 audio mixer, synthesizer등이 있다.

이러한 장비들은 하나하나가 수천만원을 상회하는 고가품으로 애니메이션을 제작하는 조직이 이 장비들을 전부 설치하기에는 부담이 너무 크기 때문에 이러한 장비들을 골고루 갖추어 놓고 전문적으로 대여해 주는 post production을 이용하는 것이 비용면에서 효과적이라고 할 수 있다. 그러나, 간단한 복제나 편집을 위하여 post production을 이용하는 것은 1일 사용료가 300만원을 육박한다는 점을 감안할 때 비경제적이라고 할 수 있다. 따라서, 간단한 편집 장비는 갖추어 놓고 비디오 합성등의 특수 효과만을 post production을 이용하는 것이 더욱 효과적이라고 할 수 있다.

최근 multi media 기기의 발달로 PC 레벨의 컴퓨터로 VTR이나 audio 기기등을 컨트롤하거나 synthesizer와 연결하여 특수 음향을 생성해 내는 시스템들이 개발되어 발표되어지고 있다. 이러한 시스템들을 이용하면 저가격으로 여러가지 특수 효과를 생성해낼 수 있을 것으로 기대되어진다.

DTPr이란 Desk Top Presentation의 약자로 Macintosh의 HyperCard등과 같은 멀티 미디어 지원 시스템을 이용하여 프리젠테이션을 위한 다양한 원고를 작성하는 시스템이다. 이 프리젠테이션 원고로는 35mm 슬라이드, OHP(Over Head Projector), 계획서 및 보고서등과 같은 인쇄물, 비디오 테이프등 프리젠테이션의 목적 및 성격에 따라 모든 미디어들이 이용되어 질 수 있다.

애니메이션의 편집 및 음악 더빙이란 결국 프리젠테이션을 위한 비디오 테이프를 작성하는 것이므로 DTPr의 일부라고 할 수 있다.

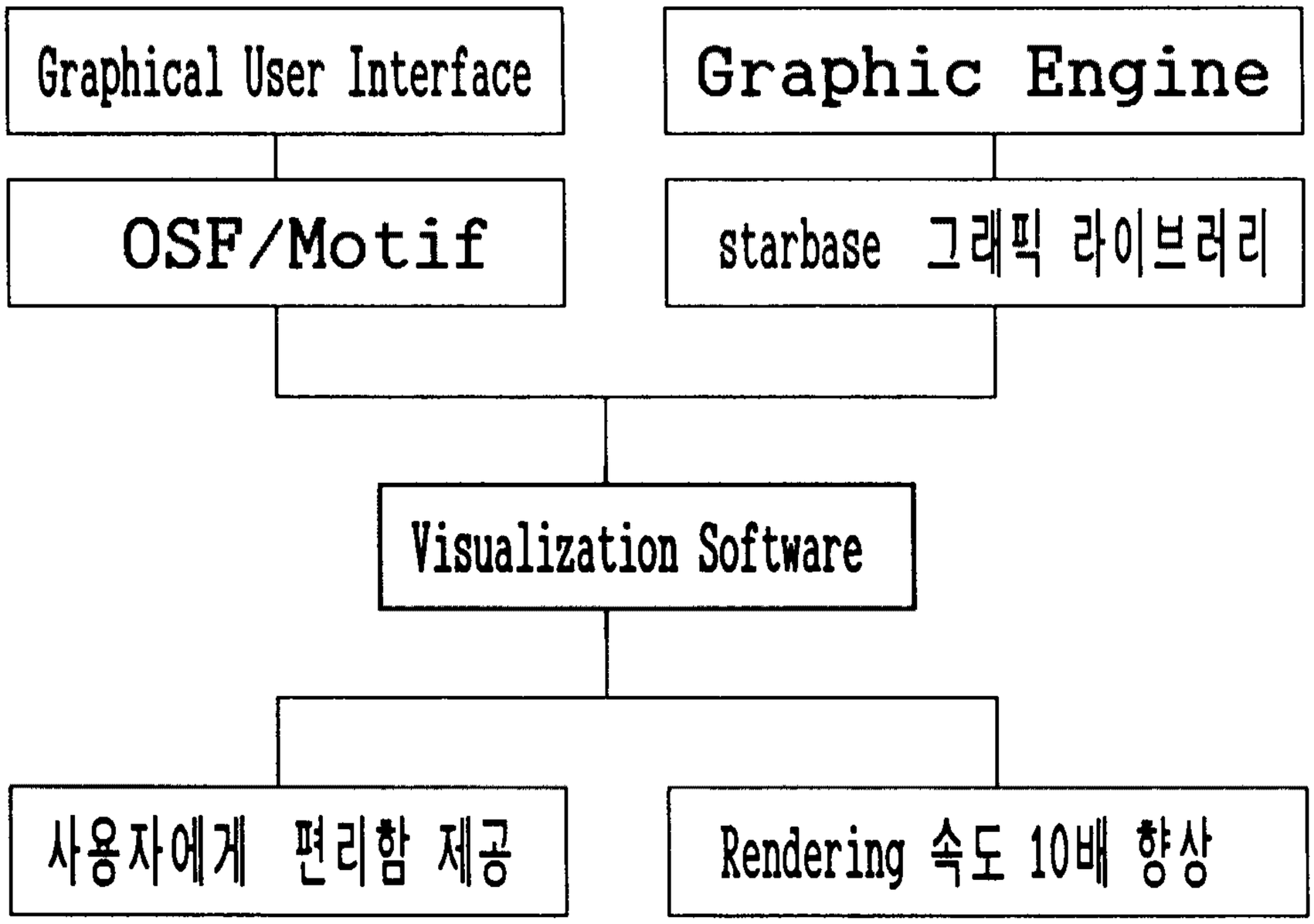
따라서, 본 연구에서는 DTPr이라는 종합적인 시스템을 구축하여 비디오의 편집 뿐만 아니라 시나리오 작성에서 각종 프리젠테이션에 이르기까지 관리할 수 있도록 한다.

제 3 장 Visualization S/W 개발

제 1 절 개 요

본 Visualization 시스템은 사용자 인터페이스(User Interface)를 중요시 하여 개발되었다. 이는 현재의 소프트웨어 개발 방향이 Graphical User Interface(GUI)를 충분히 이용하고, 향후에는 Virtual Reality를 이용한 3차원 사용자 인터페이스로 가고 있기 때문이다.

상업적인 기능을 가진 Visualization 소프트웨어를 개발하는 데는 방대한 인력과 자원이 요구되어진다. 따라서 적은 연구자원으로 개발하는



[그림 3-1] 연구의 방향 및 목표

본 연구과제의 결과물이 상업적으로 성공한 외국의 Visualization 소프트웨어보다 모든 면에서 우위를 갖도록하기에는 현실적으로 한계가 있다.

이와같은 이유에서 본 Visualization 시스템은 사용자가 편리하게 사용할 수 있는 환경을 제공하는것과 렌더링 속도 향상등의 두가지 측면에서 외국의 유명 소프트웨어와 비교 우위를 갖는 소프트웨어를 개발한다는 목표를 갖고 본 연구과제를 수행하였다.

렌더링 성능 향상을 위하여 외국 유명 소프트웨어를 분석한 결과 기존의 외국의 소프트웨어의 경우는 하드웨어 간의 호환성을 고려하여 그래픽 워크스테이션이 갖고 있는 그래픽 엔진의 성능을 활용하지 않고 대부분의 중요한 렌더링을 CPU의 성능만으로 해결하고 있다.

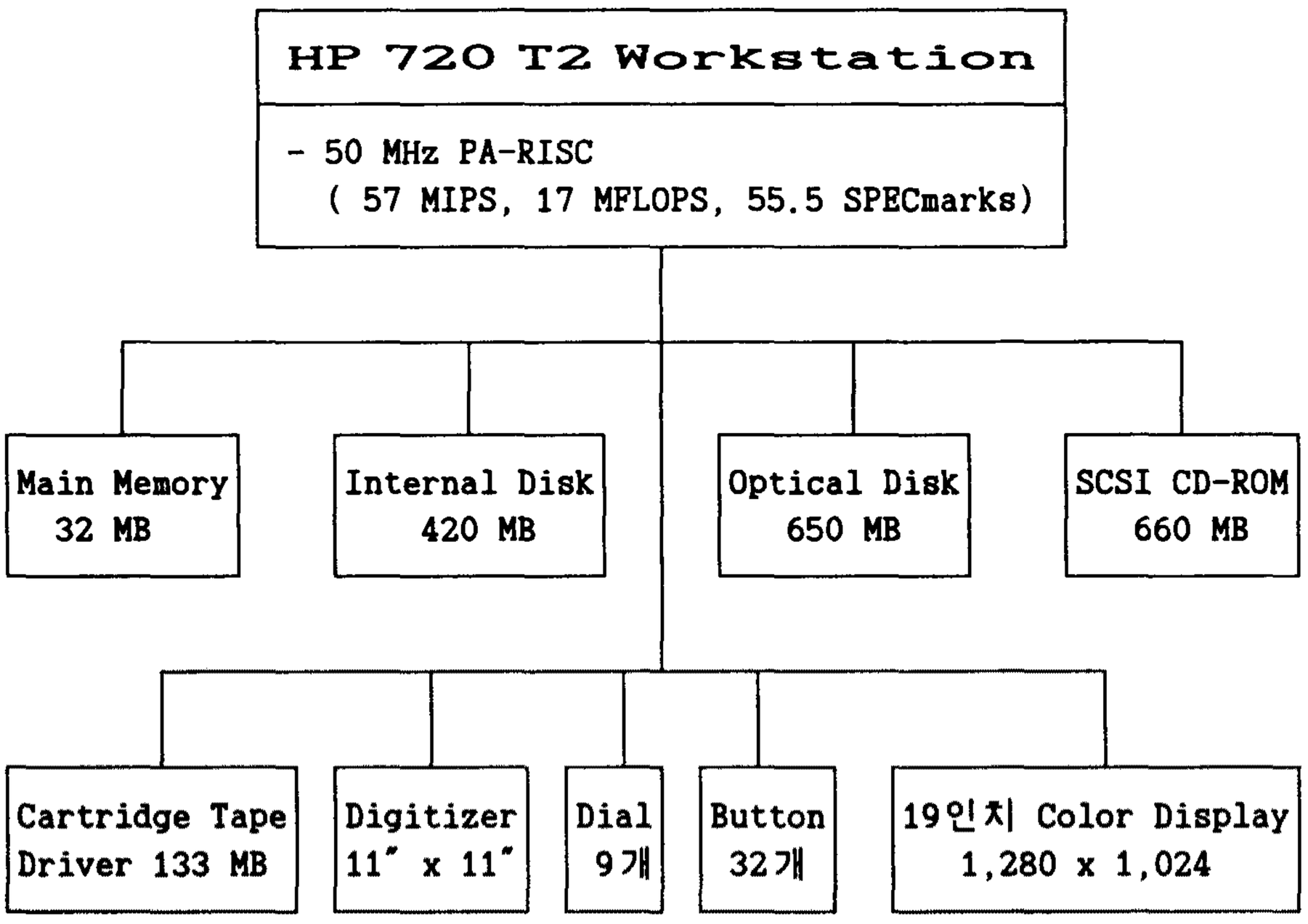
따라서 본 연구에서는 처음부터 기존의 외국 소프트웨어가 가지고 있는 범용성에서 탈피하여 우선적으로 건축분야에의 활용에 촛점을 맞추어 시스템 설계를 하였고, 그래픽 워크스테이션의 그래픽 엔진의 성능을 최대한 발휘할 수 있는 프로그램을 작성하였다. 그 결과 기존 소프트웨어 렌더링 성능의 10배 이상의 속도로 렌더링할 수 있는 시스템을 구축할 수 있었다.

최근의 컴퓨터 그래픽 분야의 기술발전 주기는 6개월 이하로 빨라지고 있다. 컴퓨터 하드웨어의 발달 또한 앞으로 2-3년내에 1000MIPS의 CPU가 개발되어질 것으로 예상되어지고 있다. 이러한 환경의 변화에 대응하기 위해서는 컴퓨터 그래픽 분야의 표준화 경향과 하드웨어 간의 호환성을 고려한 시스템 개발을 하지 않으면 안된다.

본 연구에서 사용한 그래픽 라이브러리인 starbase는 비록 그래픽 표준은 아니지만 현재의 그래픽 표준인 PHIGS 개발의 기본이 된 CGM (Computer Graphics Metafile)에 준거하여 작성된 라이브러리이며 HP사에서도 Starbase 와 PHIGS를 통합시킨 새로운 라이브러리를 개발하여 곧 공개할 예정이다. 따라서 새로운 그래픽 라이브러리가 등장하더라도 기본 구조의 변경없이 시스템을 재구성할 수 있을 것이다. 또한 본 연구에서 활용한 Graphics User Interface Tool인 OSF/Motif는 워크스테이션 분야에서의 X-window프로그램 개발을 위한 표준 tool로 정착 되어 가고 있고 PHIGS와 X-window를 병합시킨 PEX (PHIGS Extention to X-window)가 새로운 표준화 안으로 결정되어 각사에서 implementation 작업을 진행중이다. 따라서 본연구에서 사용된 그래픽 라이브러리나 Graphics User Interface는 차세대 표준화 안을 최대한 수용할 수 있는 것으로서 앞으로의 시스템 환경 변화에 충분히 적응할 수 있을 것으로 사료된다.

제 2 절 시스템 구성

본 Visualization 시스템은 [그림 3-2]와 같이 하드웨어를 구성하였다. 본체는 55.5 SPECmarks, 57 MIPS, 17 MFLOPS의 성능을 갖는 50 MHz PA-RISC 프로세스를 장착한 HP 9000시리즈 모델 720을 사용하고, 메모리는 32 MB, 내장 디스크는 420 MB, 외장 광 디스크는 양면에 650 MB, 660 MB의 SCSI CD-ROM, 테입 1개에 133 MB을 저장할 수 있는 1/4인치 카트리지 테입 드라이브 등을 연결하여 방대한 량의 이미지 데이타의 처리 및 보관에 불편이 없도록 하였다. 이외에도 11" x 11" 디지타이저와 32개



[그림 3-2] 시스템의 H/W 구성

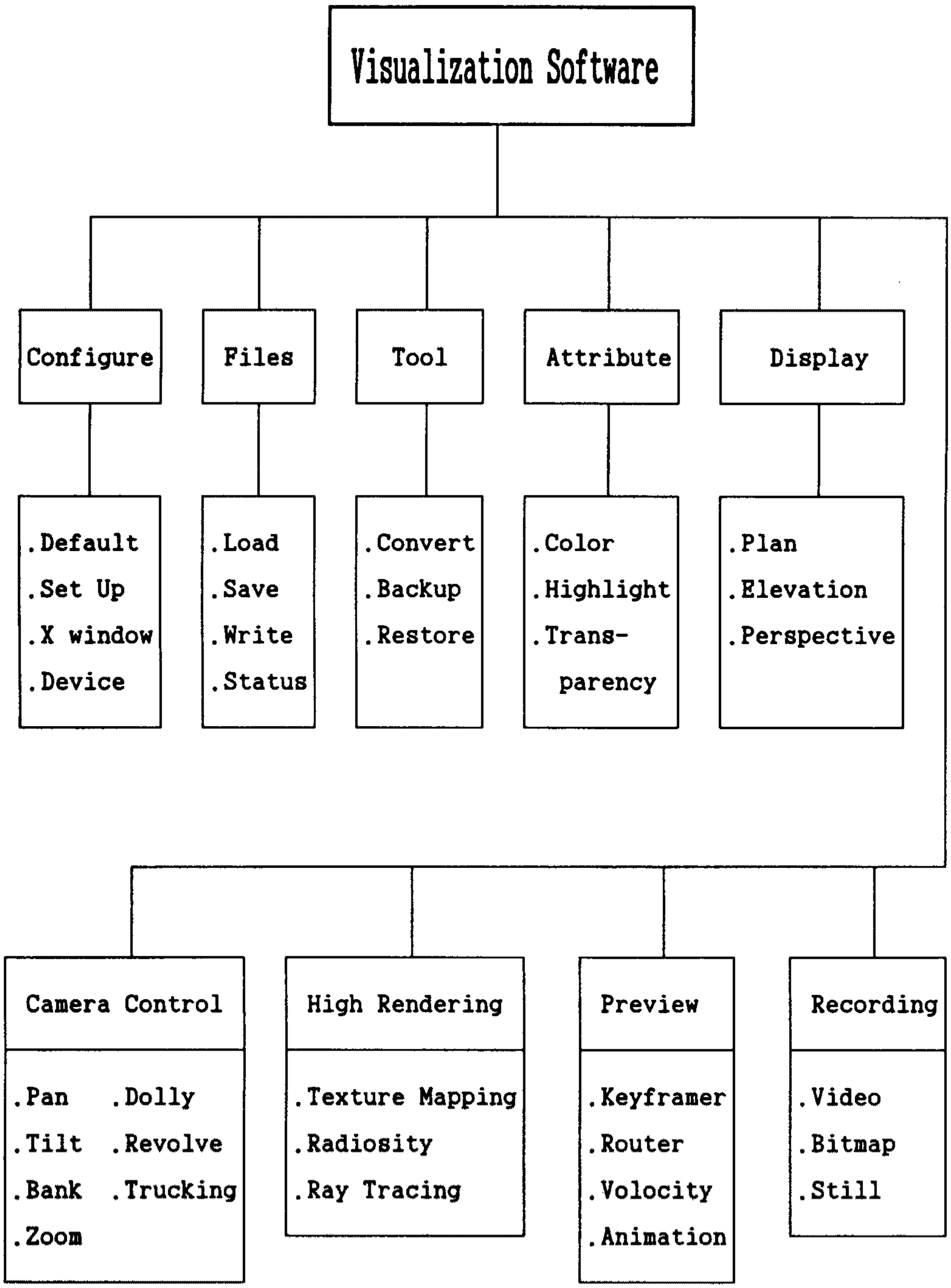
의 버튼을 갖고있는 버튼-박스, 9개의 다이알을 갖는 다이알-박스, 1,280 x 1,024의 화소를 출력하는 19인치 칼라 디스플레이로 구성된다.

본 Visualization 소프트웨어의 모듈별 구성은 [그림 3-3]과 같이 구성하였다.

Configuration 모듈에서는 Device Management를 위한 모듈로써 기본적으로 시스템을 가동할 수 있도록 하는 Default Setting 기능과 Default 값을 변경하여 Setting하는 기능이 있으며, 또 GUI의 기본인 X 윈도우의 그래픽 윈도우, 마우스, 9개의 다이알을 Setting 할 수 있는 다이알 박스, 32개의 버튼이 Setting 가능한 버튼 박스등의 디바이스들을 사용자의 편의에 따라 변경 작동 가능하도록 했다.

Files 모듈은 Geometric 데이타, Light 데이타 등의 작업에 필요한 자료들을 Load, Save, Write 한다. 또, 메모리에 Load 되어 있는 자료들의 크기, 갯수, 이름 등을 볼 수 있도록 하는 메모리 Status 기능이 있어서 메모리의 활용 효율을 높히기 위한 시도를 하였다. 즉, 불필요한 자료가 메모리에 존재 할 경우에는 이를 메모리에서 지움으로써 꼭 필요한 자료만으로 작업을 이루워지도록 하여 메모리를 절약하고 작업의 신속성을 높혔다.

Tool 모듈은 PC에서 작업된 형상 모델링 자료들을 DXF 화일 형식에 의하여 렌더링 작업에서 사용할 수 있는 자료로 변환하는 작업과 TIPS와



[그림 3-3] 시스템의 S/W 구성

같은 2차원 페인팅 시스템에서 채취한 질감자료들을 변환하는 작업을 하는 기능을 갖는 Convert 모듈과 방대한 량의 이미지 데이타를 작업 도중에 Backup를 할 수 있는 기능을 부여하여 Internal 디스크를 효율적으로 관리 할 수 있도록 했다.

Attribute 모듈은 Diaplay 모듈에서 나타내는 Object 자료의 색상을 조절하는 Coloring 기능과 각각의 Object의 Transparency를 조절하여 Fade In, Fade out 의 효과를 줌으로써 서서히 나타나게하거나 서서히 없어지도록 할 수 있다.

Display 모듈은 Camera Control 모듈과 함께 1차년도 연구의 핵심 모듈이다. 메모리에 Load 된 Object를 원하는 시점에서 3차원으로 자유자재로 볼 수 있는 Viewing기능이 있다. 해당 Obejct의 Top View, South View, East View 등의 3개 방향의 View와 Perspective를 동시에 볼 수 있도록 한다. 이와같은 Viewing 기능 외에도 각각의 부재를 Allocation 하여 부재의 조립과정등을 시뮬레이션 할 수 있도록 하였다.

Camera 모듈은 카메라의 조작을 손쉽게 할 수 있도록 Pan, Dolly, Tilt, Revolve, Bank, Trucking, Zoom 등의 기능을 부여하였다.

High Rendering 모듈, Preview 모듈, Recording 모듈은 2,3차년도의 연구과제로써 1차년도에는 기본설계만 완료하였다.

제 3 절 입.출력 자료구조

본 Visualization 시스템에서 사용하는 자료들은 구조를 최대한으로 단순화 시킴으로써 타 시스템에서 개발한 자료들을 손쉽게 변경하여 사용할 수 있도록 하였다. 자료는 Binary화일로 Write하여 저장 용량을 줄이고, 메모리에 Load 하는 시간을 단축시키는 방법을 이용했다.

물체의 형태와 모양을 나타내는 형상 자료는 [표 3-1]과 같이 3차원 좌표값을 갖는 P(x,y,z)과 단순이동이나 라인을 그리느냐의 Control id 만을 갖고 있으며, Wire Frame(.wir) 모델이나 Constant Shading(.cst) 모델 어느것에도 같이 사용한다. 다만 Constant Shading(.cst)모델은 시작점과 끝점이 일치해야 평면을 이룰수 있기 때문에 Wire Frame(.wir) 모델과 차이가 있다.

[표 3-1] 형상 정의를 위한 기본 자료구조(.wir)(.cst)

```
struct {
        float   x;    /* X,Y,Z의 3차원 좌표값 */
        float   y;
        float   z;
        float   id;   /* Move (0), Draw (1)   */
} Wire_data;
```

Constant Shading은 일정한 밝기로 다각형을 칠하는 가장 단순한 방법인데 반하여 Gouraud Shading은 다각형의 법선을 계산하여 밝기나 색을

보간하는 방법이므로 [표 2-2]와 같은 법선(Nomal Vector)에 대한 정보 NP(nx,ny,nz)가 추가로 포함되어야 한다.

[표 3-2] Grouraud Shading을 위한 자료구조(.nrm)

```
struct {
        float   x;    /* X,Y,Z의 3차원 좌표값 */
        float   y;
        float   z;
        float  nx;    /* 법선의 3차원 좌표값  */
        float  ny;
        float  nz;
        float  id;    /* Move (0), Draw (1)   */
} Grouraud_data;
```

지형을 3차원으로 표현하기 위해서 삼각형 Patch의 연결로 표현하는 데 이에 사용하는 (.top)자료의 구조는 [표 3-3]과 같다.

[표 3-3] Topology를 위한 자료구조

```
struct {
        float   x;    /* X,Y,Z의 3차원 좌표값 */
        float   y;
        float   z;
} Topology_data;
```

Display 모듈은 본 시스템의 기본 모듈로써 퍼스날 컴퓨터에서 작업한 모델링 자료를 워크스테이션에서 칼라링(Coloring), 뷰잉(Viewing) 작업을 하는 핵심 모듈로써 [표 3-4]와 같은 화일구조를 갖는다.

Model_Type은 Wire frame data, Constant shading data, Gourand shading data, Image data 중에서 어떤 모델 Type 인가를 알려주는 Index 역할을 한다.

Layer_No, Class_No, Segment_No는 Display를 위한 부재들을 Level별로 분리하여 나타내도록 한다. 즉, Level-1, Level-2, Level-3등의 3개의 Level로 분리하여 색깔 및 이동을 따로 따로 할 수 있도록 하여준다.

File_Name은 퍼스날 컴퓨터에서 3차원 솔리드 모델러를 이용한 모델링 작업은 부품별 또는 부재별로 나누워 모델링하여 별도의 화일로 보관되어야만 부품별 또는 부재별로 애니메이션이 가능하기 때문에 이 각각의 화일 이름을 갖는 필드이다.

RGB 필드는 물체의 색상을 표현하는 칼라링 기능을 갖도록 한다.

Shade_Switch 필드는 광원의 영향이 있도록 할 것인가 광원과 무관하게 Display 할 것인가를 결정하는 필드이다.

Visibility_Switch는 Visibility 작업을 할 것인가 안할것인가를 표시하는 필드이다.

Highlight_Switch,_Power,_RGB는 금속광택과 같은 효과를 내기 위하

[표 3-4] Display (.dis) 화일구조

```
struct {
  int       model_type;              /* type of model              */
  int       layer_no;                /* number of layer            */
  int       class_no;                /* number of class            */
  int       seg_no;                  /* number of segment          */
  char      f_name[MAXNAME];         /* data file name             */
  int       rgb[3];                  /* RGB of each object file    */
  int       shade_sw;                /* shading switch             */
  int       vis_sw;                  /* visibility switch          */
  int       highlight_sw;            /* highlight switch           */
  int       highlight_power;         /* highlight power            */
  float     highlight_rgb[3];        /* highlight color            */
  int       transparency_sw;         /* transparency switch        */
  int       transparency_rate;       /* transparency rate          */
  int       zbuffer_sw;              /* zbuffer on off             */
  int       move_model_sw;           /* move model or not          */
  float     rel_rot[3];*/            /* object relative rotation   */
  float     rel_loc[3];*/            /* object relative location   */
  float     abs_loc[3];              /* object absolute location   */
  float     abs_rot[3];              /* object absolute rotation   */
  float     range[2];                /* topology data range        */
  int       topo_hole_nu;            /* hole number of topo data   */
  float     topo_hole_range[10][4];/* (minx miny maxx maxy)        */
  int       maxpoint;                /* max vertex  in each file   */
  float     *vertex;                 /* vertex list arrays         */
  int       light_use;               /* light use sw [default:0]   */
  Light_data      *light;
  } Obj_data;
```

여 Highlight를 조정하는 필드이다.

Transparency_Switch,_Rate는 0 부터 15 까지의 16단계의 투명도를 갖도록한다. 0는 불투명, 15는 완전투명한 값을 갖도록 한다.

Zbuffer_Switch는 Zbuffer 알고리즘을 사용할 것 인가를 결정하는 필드로써 일반적으로 Set(1)을 갖고 Zbuffer 알고리즘을 사용한다. 그러나 지형자료와 건물의 바닥자료가 일치하는 부분에서는 미세한 거리 차이에 의한 간섭이 발생하는 문제가 발생한다. 이런 경우에는 지형 자료를 Zbuffer_Switch를 Reset(0)하여 간섭 문제를 해결한다.

Rel_Rot, Rel_Loc, Abs_Rot, Abs_Loc 필드는 대상물체의 절대좌표(Absolute)와 상대좌표(Relative)를 Location 하고 Rotation하는데 사용한다.

Range[2]는 Topology 자료의 X,Y를 나타내는 필드이다.

Topo_Hole_Number는 Topology 자료에 몇개의 Hole을 냈는가를 나타내는 필드이다. 이 필드는 Zbuffer_Switch와 함께 간섭을 배제하는데 사용되는데 이는 지형자료에 건물의 바닥자료 크기만큼의 구멍을 뚫어서 자료의 크기도 줄이고, 간섭이 생길수 있는 대상자료를 없앤다.

Vertex 필드는 물체의 형상자료를 갖고있는 포인터이다.

Light_Use Switch는 사용되는 Light의 번호를 갖고 있어서 Light 화일의 정보를 사용할 수 있도록 한다.

Lighting(.lgt)화일은 광원을 정의하는데 사용한다. 광원의 타입은 Directional Light와 Positional Light으로 구분하여 사용한다. 광원의 색상은 RGB 값으로 표현하고, 광원의 위치와 조명이 물체를 비추는 벡터 값을 갖도록 하고, Attenuate Light, Spot Light, Cone Light의 성질을 포함하여 성질에 따라 광원의 영향을 달리하는 처리를 하였다.

[표 3-5] Lighting (.lgt) 화일구조

```
struct {
  int   use_no;      /* light set number                              */
  int   light_no;    /* total number of light in one set              */
  int   type[MAX];   /* light type [DIRECTIONAL:0,POSITIONAL:1]       */
  int   red [MAX],
        green[MAX],
        blue[MAX];   /* color of light [0-255]                        */
float locx[MAX],
        locy [MAX],
        locz[MAX];   /* location of light source                      */
float tarx[MAX],
        tary[MAX],
        tarz[MAX];   /* target vector from light source               */
int   attr[MAX];   /* light attribute [ATTENUATE_LIGHT:2,
                                  SPOT_LIGHT:4,CONE_LIGHT:8]          */
int   spot[MAX];   /* spot light power [1-16383]                    */
float atten[MAX]; /* attenuation factor [0.0-1.0]                   */
float angle[MAX]; /* angle of cone light [DEGREE]                   */
int   on_off[MAX];/* light on off [ON:1,OFF:0]                      */
} Light_data;
```

Eyepoint(.eye)화일은 관찰자의 시점(Eye Point)과 대상물의 참고점(Reference Point)에 관한 정보를 갖고있는 화일이다. 계산 시간과 오류를 줄이기 위하여 Eye Point 앞을 Clipping 하고, 대상물체의 앞과 뒤를 Clipping하여 사용한다.

Camera Control을 위한 자료들을 포함하게 하고, Projection 필드를 갖도록하여 Perspective로 표시할 경우와 Parallel하게 투영하여 Axonometric으로 나타낼 경우로 분리하여 표시되도록 한다.

[표 3-6] Eyepoint (.eye) 화일구조

```
struct {
  float  eye[3];       /* eyepoint x,y,z                              */
  float  ref[3];       /* reference point x,y,z                       */
  float  vf;           /* front clipping distance from refpoint       */
  float  vf_safety;    /* front clipping distance from eyepoint       */
  float  vb;           /* back clipping distance from refpoint        */
  float  focus;        /* focus value of camera                       */
  float  angle;        /* angle of viewing cone for HP                */
  float  bank;         /* camera banking (unit DEGREE)                */
  float  upvector[3]; /* camera up vector                             */
  int    projection;   /* projection type                             */
} Eyepoint_data;
```

Visibility(.vis) 화일은 Display (.dis) 화일에 정의된 Layer_No, Class_No, Seqment_No에 해당되는 물체들을 Start_Cut부터 End_Cut까지

의 화면에 보이게 하여 일종의 Animation을 할 수 있도록 한다.

[표 3-7] Visibility (.vis) 화일구조

```
struct {
  int  layer_no;    /* Level 1 의 분류 : Layer   */
  int  class_no;    /*       2         : Class   */
  int  seg_no;      /*       3         : Segment */
  int  start_cut;
  int  end_cut;
  } Visibility_data;
```

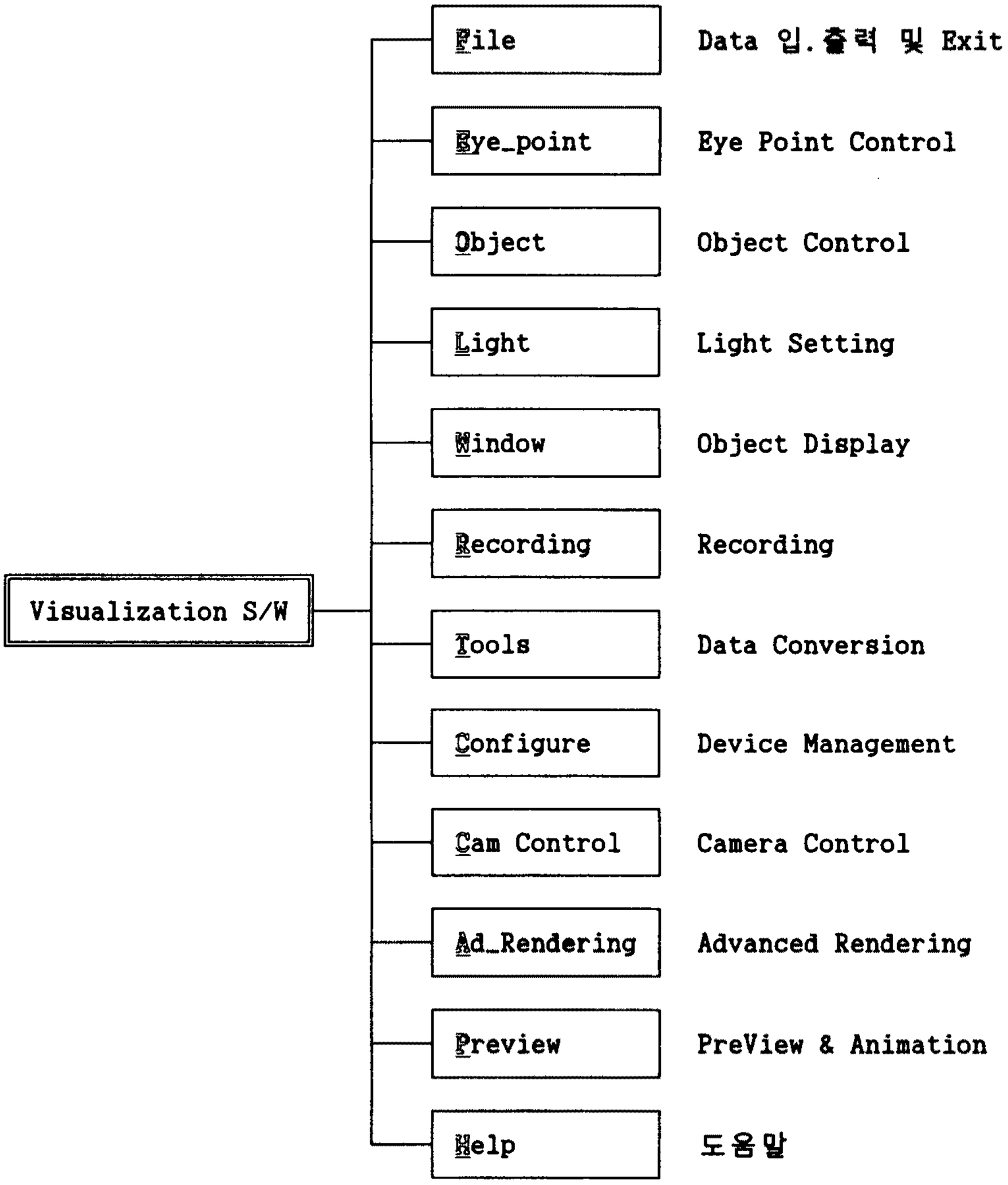
Object Move를 위한 (.mov) 화일구조는 Visibility(.vis) 화일과 유사한 구조를 갖는다. 물체가 이동하는 점들을 포인터로 갖고 있도록 하여 이동을 위한 Move Curve를 나타내어 Animation을 할 수 있도록 한다.

[표 3-8] Object Move를 위한 (.mov) 화일구조

```
struct {
  int  layer_no;    /* Level 1 의 분류 : Layer   */
  int  class_no;    /*       2         : Class   */
  int  seg_no;      /*       3         : Segment */
  int  start_cut;
  int  end_cut;
  float *vector;    /* Move  Vector              */
  } Move_object;
```

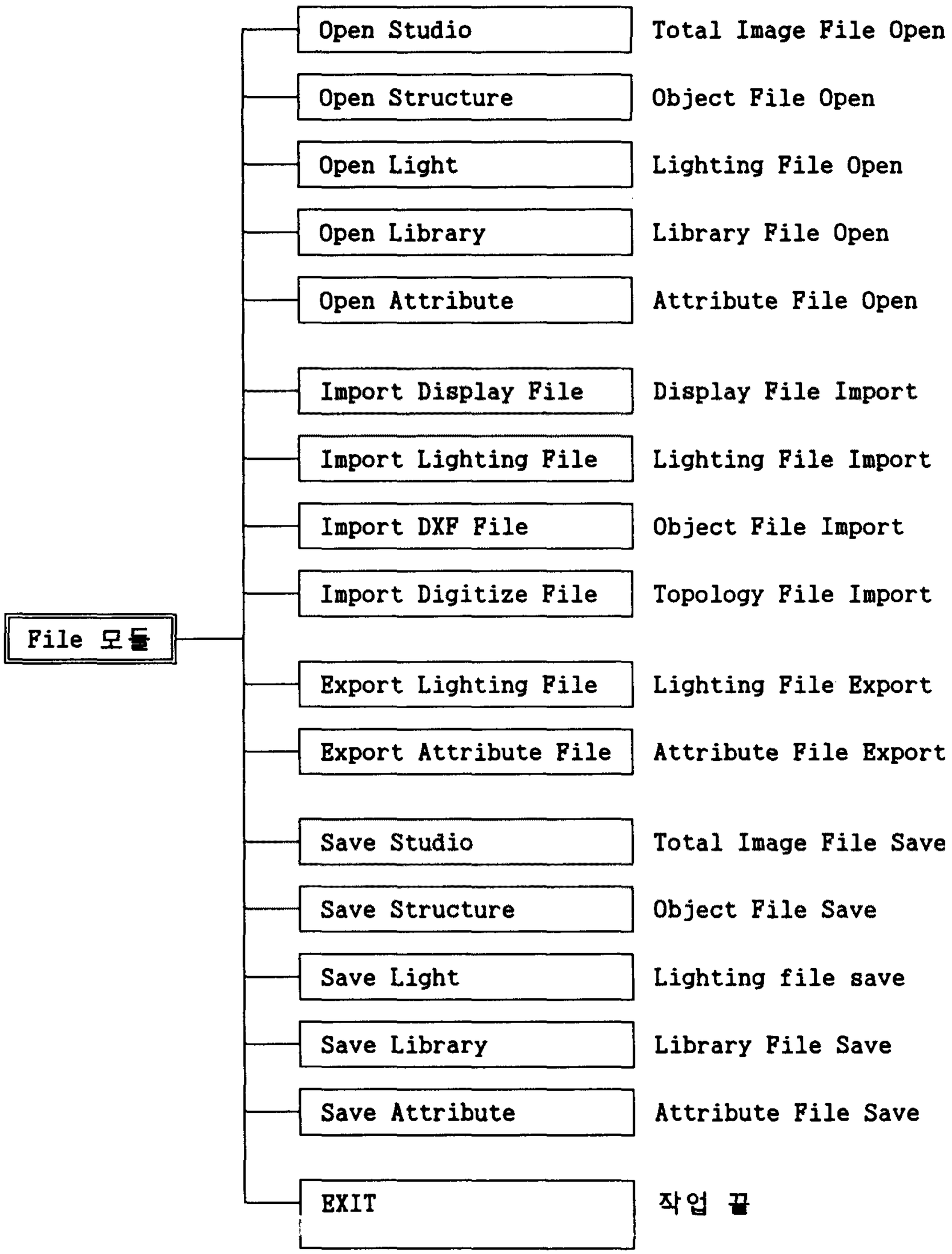
제 4 절 모듈별 기능

본 Visualization 시스템의 모듈구성은 [그림 3-4]와 같이 12개의 모듈로 이루워졌다.



[그림 3-4] 시스템의 Top Level 메뉴에 의한 모듈 구성

1. File 모듈



[그림 4-5] File 모듈의 기능

File 모듈은 Visualization S/W를 동작시키기 위한 필요한 자료들을 Open, Import, Export, Save하도록 하는 기능을 갖고 있다.[그림 3-5]

가. Open, Save

Open 기능은 화일을 읽어서 메모리에 Load 시키는 기능을 말하고, Save 기능은 메모리의 자료를 Disk에 Write하는 기능을 말한다.

Studio는 방송용어로 주로 사용되는 단어로써 본 Visualization S/W에서는 Display를 위한 최종 이미지 단위를 말한다. 즉, 하나의 Studio를 설치하기 위해서는 조명을 어떻게 설치할것이며, Object들을 어떻게 배치할것이며, 각각의 조명과 Object들은 어떤 색상을 갖도록 하는것이 실제감을 더낼 수 있을까 등등을 고려하여야한다. 이와같은 고려사항에 착안하여 만들어진 이미지 자료를 Studio 자료라고 말한다.

Structure 자료는 Object들의 Hierachy를 나타내는 자료를 포함하는 형상자료를 말한다. 즉, 일부의 S/W에서 사용하는 Group과 비슷한 의미를 갖고 있다. 의자라는 Structure 자료는 4개의 의자 다리 형상자료와 방석이 위치하는 널판지와 등받이 등으로 구성되는 Structure 자료이다.

Light 자료는 광원에 관한 정보를 포함하고 있다. 광원은 형상자료의 색상과 그림자등에 직접적으로 변화를 일으키기 때문에 광원의

타입에 따라 미리 만들어 영향에 대한 분석을 해 놓는 작업을 거친후 필요할때는 Open Light 만을 하여 사용할 수 있도록 하고 있다.

Library 자료는 형상정의 자료로써 부재라고 부르는 부품의 최소 단위부터 부품까지를 Library에 등록하여 두고 필요할때 마다 불러서 사용하는 방법을 택하고 있다. 즉, 물품 창고라고 생각하면 된다. Studio에 의자가 필요하면 창고에서 원하는 타입의 의자를 꺼내서 배치만 하면 되도록 하였다.

Attribute 자료는 Color, Highlight, Transparency등의 기능을 이용하여 형상 자료를 변환하여 원하는 효과를 내는데 사용되는 자료이다.

나. Import, Export

Import는 다른 S/W 또는 다른 모듈에서 자료를 받아들이는 기능을 말하고, Export는 다른 S/W 또는 다른 모듈로 자료를 보내는 기능을 말한다.

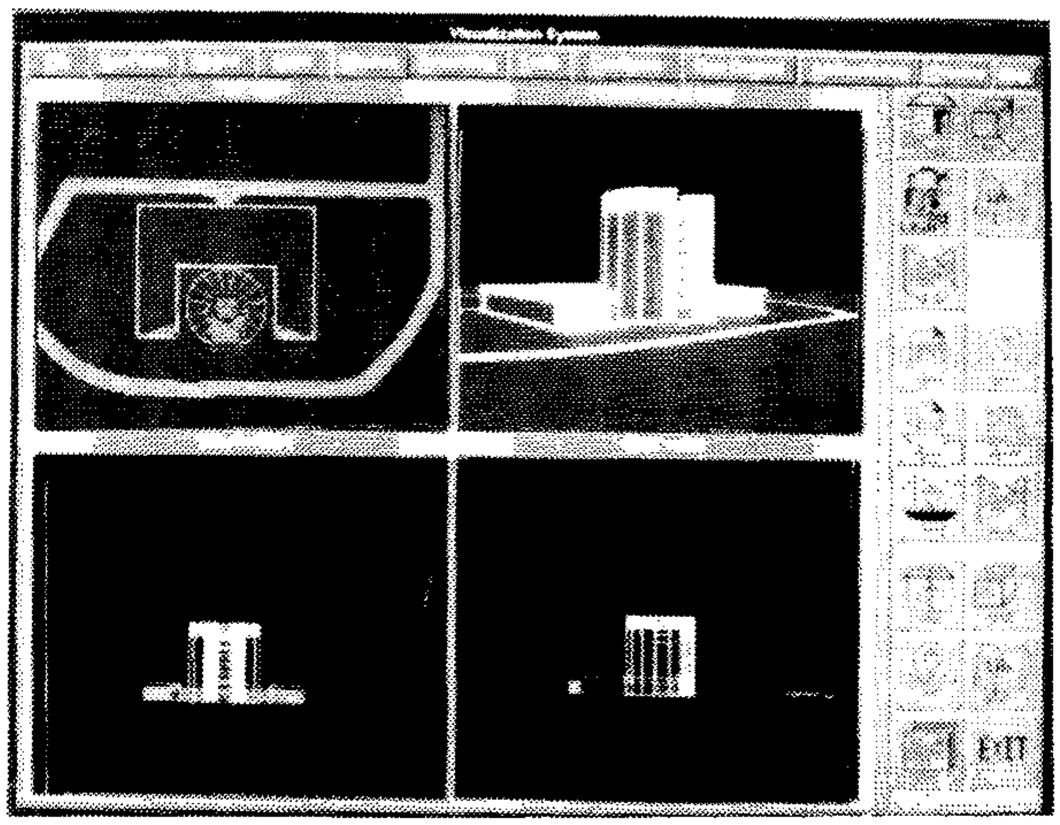
형상자료를 정의하는 작업인 Modeling 작업은 퍼스날 컴퓨터에서 3차원 솔리드 모델러인 AutoCAD r.11 AME에 의해서 작업이 이루어진다. 이때 만들어진 화일(.dwg)를 본 Visualization S/W에서 이용하기 위해서는 DXF (Data eXchange File)타입에 의해 자료를 받아들임으로

써 모델링 모듈과 연계를 갖는다.

지형자료는 Digitizer에 의하여 좌표값을 입력을 받은 후 이를 본 Visualization S/W에서 필요한 Topology 자료 형태로 변경하여 사용한다.

다. Exit

Open 되어있는 모든 Device, File들과 Save 안한 자료들을 찾아서 Save 할것을 물어보고 사용자가 원하는 작업을 수행하고 본 시스템의 동작을 끝낸다.

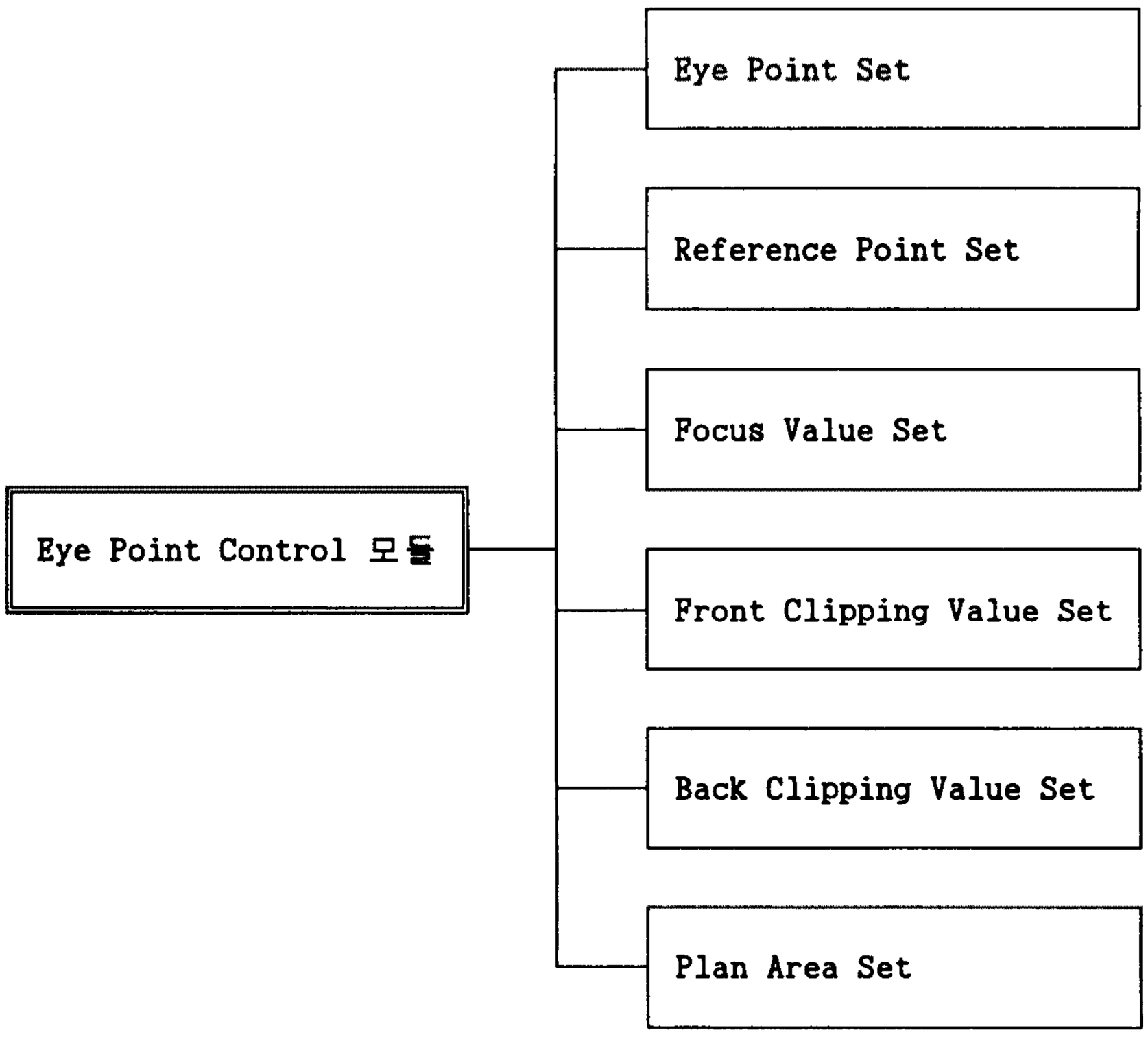


[그림 3-6] File 모듈의 화면

2. Eye Point Control 모듈

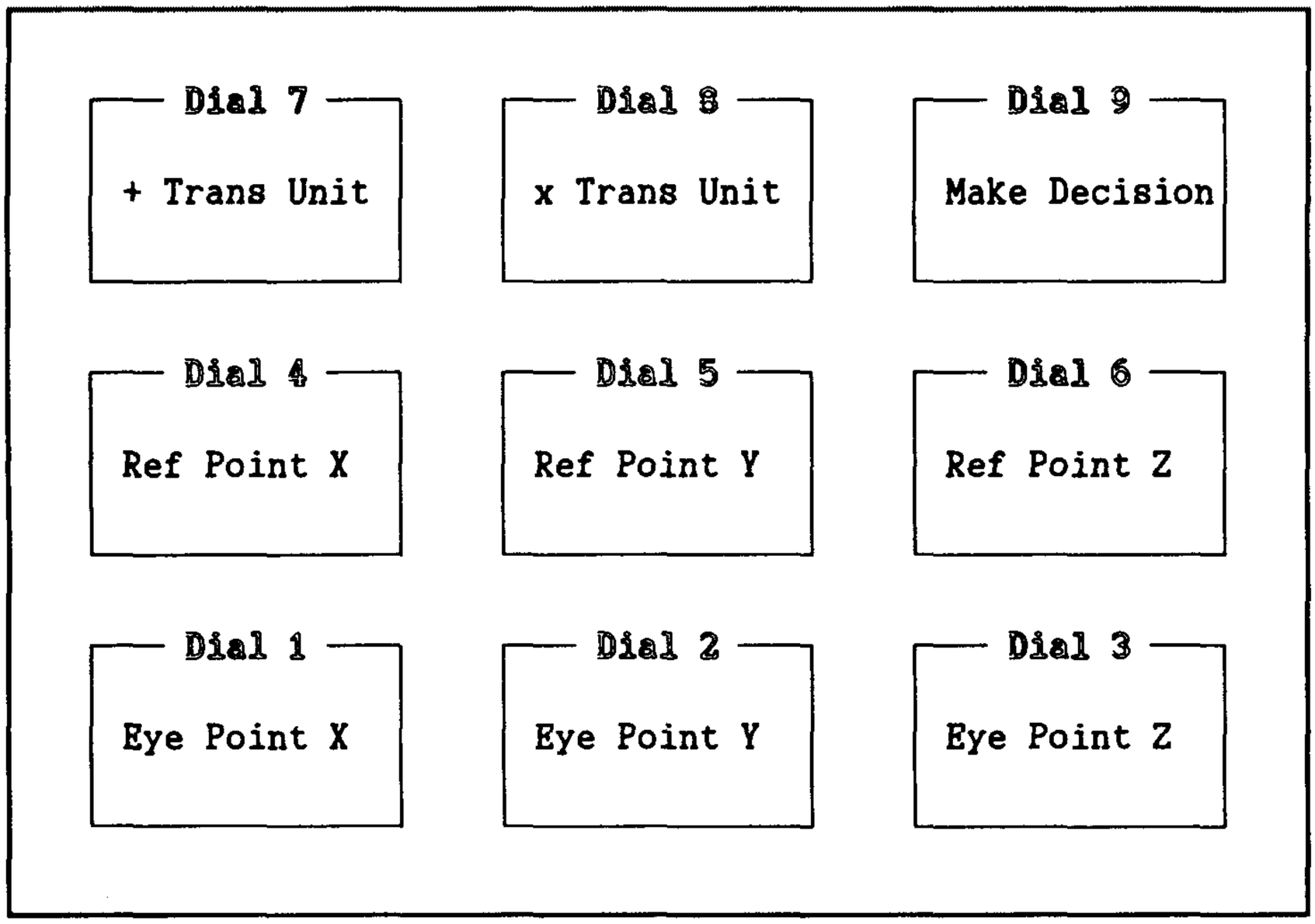
가. Eye Point와 Reference Point

관찰자의 시점(Eye Point)과 물체의 참고점(Reference Point)의 변경은 Visualization S/W의 대표적인 기능인 Viewing과 직접적인 관계를 갖고 있기 때문에 Eye Point의 변경을 사용자가 불편없이 자유자재로 변경이 가능하도록 하여야한다.



[그림 3-7] Eye Point Control 모듈의 기능

본 시스템에서는 [그림 3-8]과 같이 9개의 Dial에 각각의 기능을 부여하여 Eye Point와 Reference Point가 편리하게 변경 되도록 하였으며, 임의의 좌표를 입력할 수 있는 방법에도 마우스로 Scroll Bar를 Pick 하여 좌표값의 입력을 대신 할 수도 있게 하였다.



[그림 3-8] 9개 Dial의 Eye Point Control 기능

나. Camera의 Focus Value

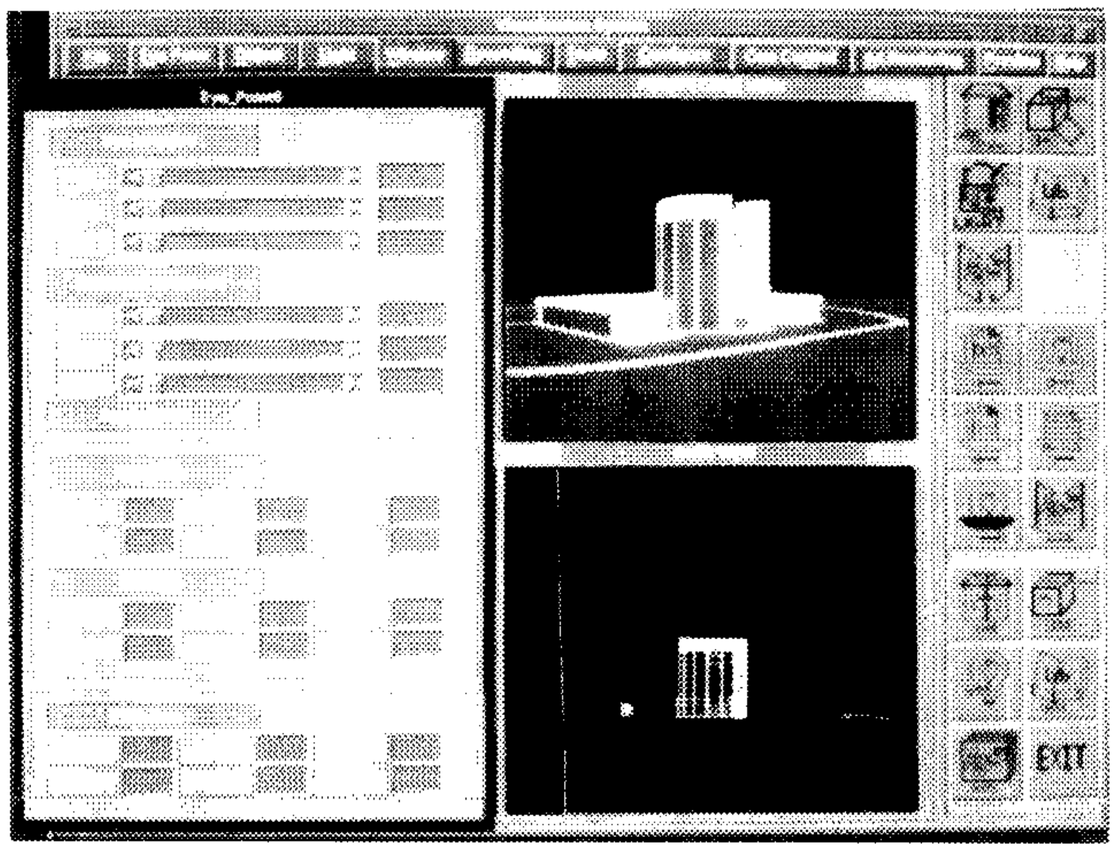
카메라의 초점거리(Focus Value)는 Eye Point와 Reference point의 변경과 함께 Viewing 모들의 중요한 요소이다. 카메라의 Zoom 기능에 직접적으로 영향을 주는 초점거리는 값이 커짐에 따라 Zoom In, 작아짐에 따라 Zoom Out된다.

다. Clipping

Clipping은 보고자하는 물체를 중심으로 전면과 후면을 자르는 것으로 Front Clipping, Back Clipping이 있다. 이와같이 Clipping 하는 이유는 2가지가 있는데, 첫번째는 좌표값이 Device Coordinate 보다 큰 값을 갖게될때 좌표값이 Truncate 되어 좌표가 변경되는 오류를 범하기 때문에 프로그램 내에서 자동으로 Back Clipping을 함으로써 Truncate Error를 배제하도록 한다. 두번째 이유는 화면에 물체를 표시하기 위해서는 해당되는 Pixel에 점멸로써 나타내는데 Back Clipping 값과 Front Clipping 값의 차이가 너무 크게 나타나면 한 Pixel에 해당되는 물체가 차지하는 영역이 커지게 된다. 이와같은 현상이 심하면 서로 다른 물체가 1 Pixel에 함께 표시되는 경우가 발생하여 물체의 윤곽선이 정확하게 표현되지 않는다. 이런경우에는 Back Clipping 값을 줄여서 지정함으로써 해결될 수 있다.

라. Plan Area

화면에 나타난 Total Image 중에서 보고자하는 특정 부분을 확대하여서 볼 수 있도록 하는 기능이다.



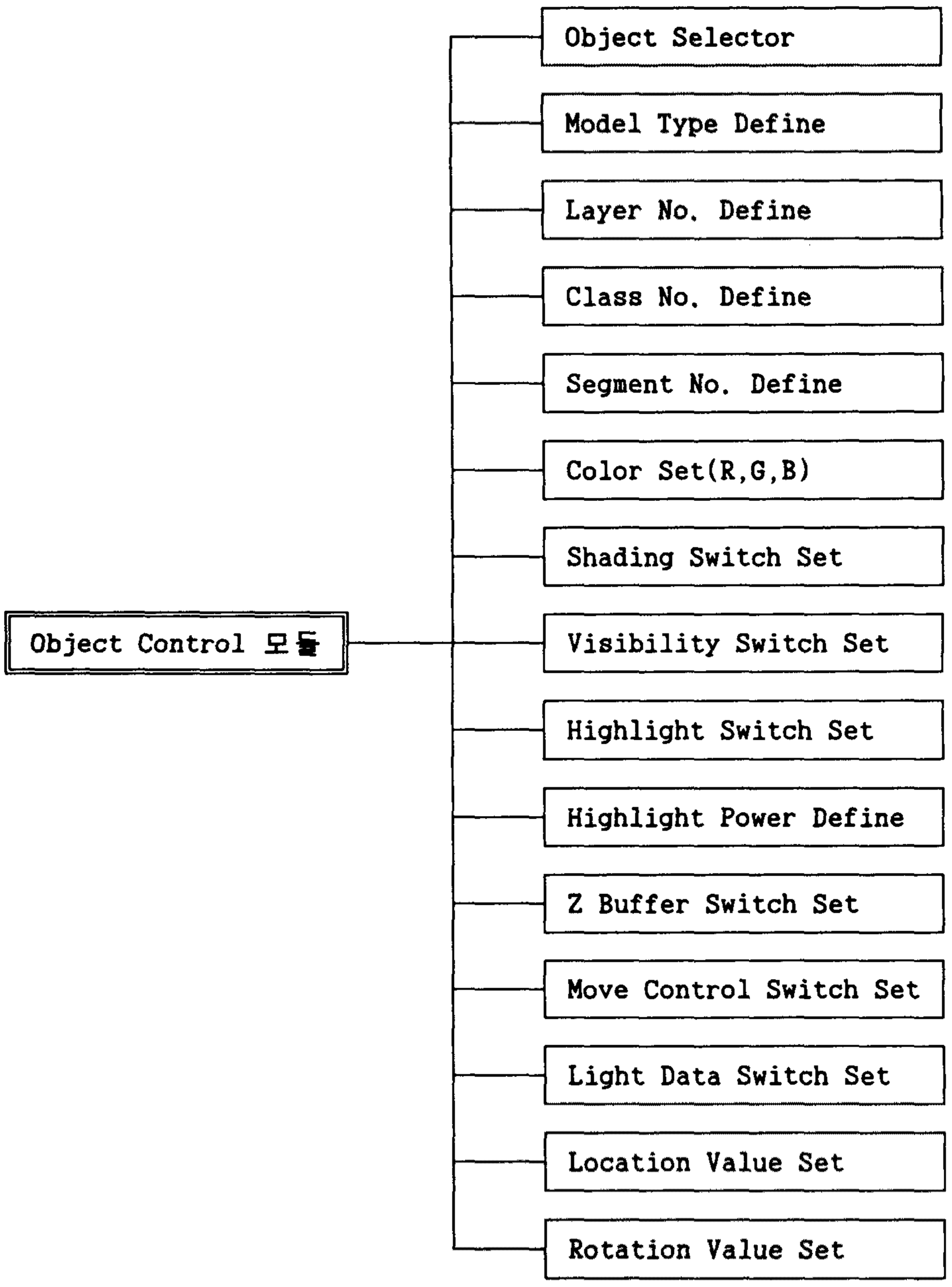
[그림 3-9] Eye Point Control 모듈의 화면

3. Object Control 모듈

가. Object Selector

하나의 화면을 구성하는 Object의 갯수는 적게는 몇 십개에서 많게는 몇 백, 몇 천까지 된다. 이렇게 많은 Object들의 이름을 모두 다 정확히 기억하기란 용이하지 않다. 해당의 Object들을 필요할때마다 빠른시간에 선택할 수 있는 방법이 모색되어야 한다. 본 시스템에서는 Scroll Bar에 의한 Object Select 방법과 CodeNo.를 직접 입

력하여 선택하는 2가지 방법을 병행할 수 있도록 하였다.



[그림 4-10] Object Control 모듈의 기능

나. Object 정보 Edit(Model Type, Layer, Class, Segment)

모든 Object는 Model Type과 Layer No., Class No., Segement No. 등을 갖고 있으며 이와 같은 정보는 필요에 따라 수시로 변경하여야 하기 때문에 편리하게 변경될 수 있도록 일종의 Editor기능을 갖도록하여 Object 정보에 대한 편리한 변경을 가능하게 하였다.

다. Color Set(R,G,B)

Coloring 기법은 컴퓨터그래픽스 시스템의 근간을 이루는 기본기술로써 R,G,B 값의 조합으로 1,670만색을 만들어 낸다. 색상에 대한 감각과 경험이 적은사람의 경우에도 Color의 R,G,B 값을 쉽게 선택할 수 있도록 인터렉티브하게 R,G,B의 값의 조합을 색상으로 보여주는 방법을 택하였다.

라. Switch Set

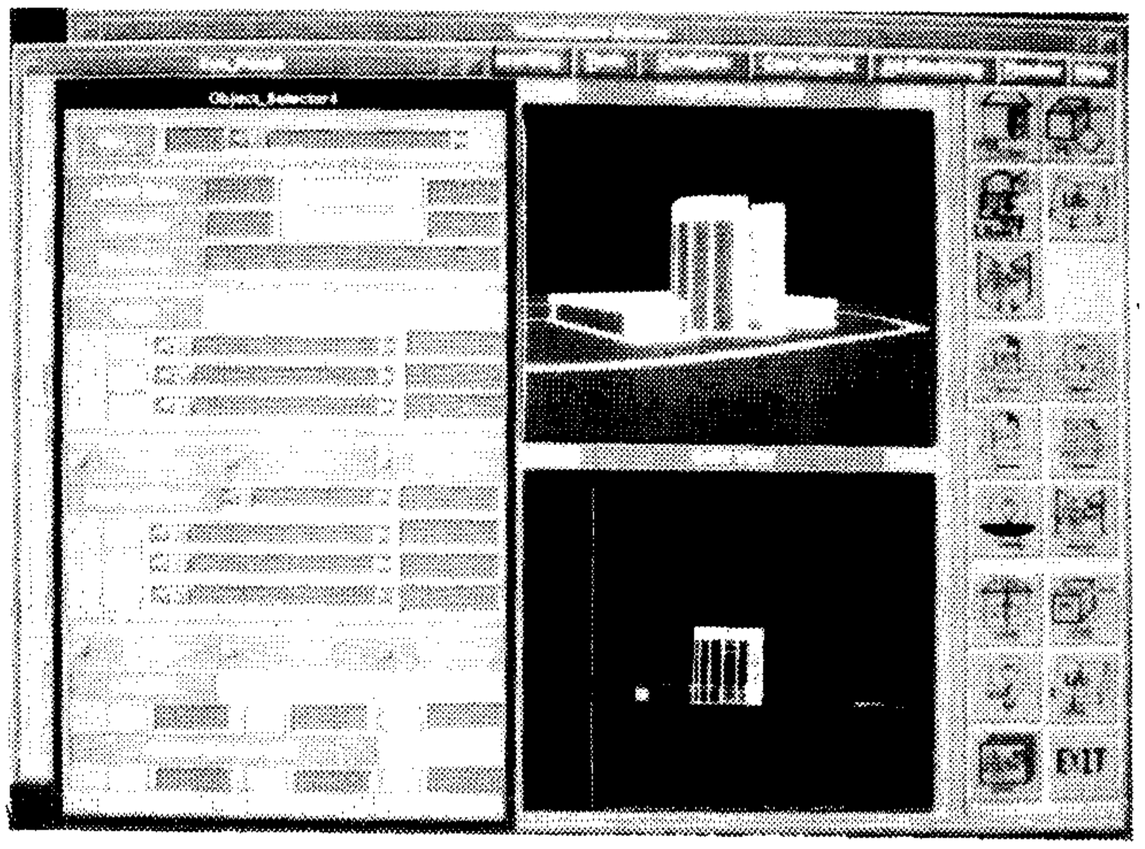
Shading 기능, Visibility 기능, Highlight 기능을 갖도록 하고 이와 같은 기능을 필요시 작동되도록 Switch를 설치하였다. 좀 더 빠른 프로그램의 수행과 기능의 작동의 유.무의 선택을 위하여 Z buffer 알고리즘 실행 유.무 기능, Light Set 유.무 기능을 Switch로 처리함으로써 프로그램 내부에서 필요이상의 IF Check를 피하였다.

마. Highlight Power Define

Object에 비추는 Highlight에 대한 정도를 표시한다. 1 - 16,383 사이의 값으로 나타내는데 최대로 Highlight되는 값은 1 이다.

바. Object 위치 변경

Object의 위치변경을 위하여 Location, Rotation 값을 Setting 할 수 있도록 하여 Object를 자유자재로 Control 할 수 있도록 하였다.

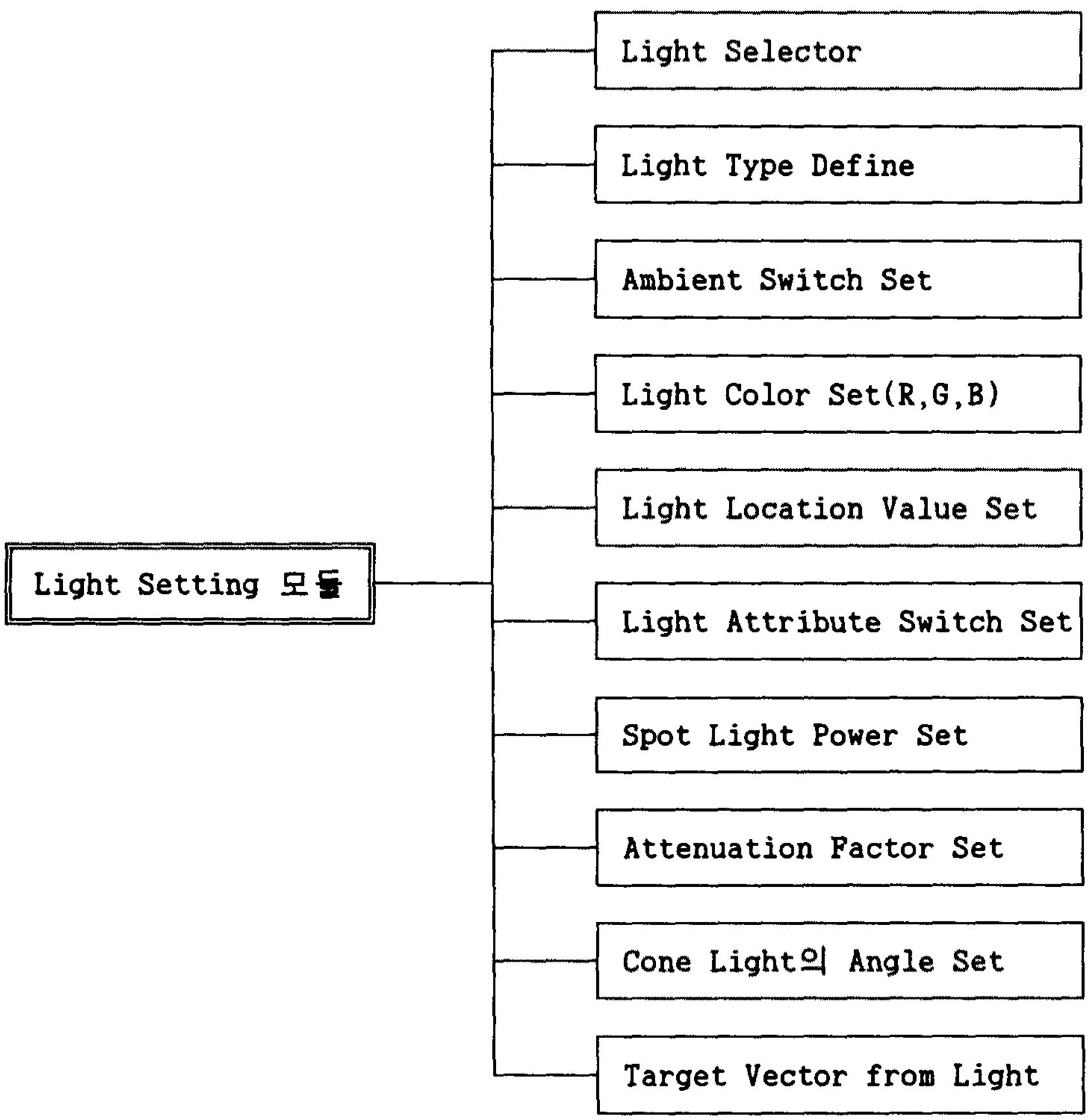


[그림 3-11] Object Control 모듈의 화면

4. Light Setting 모듈

가. Light Selector

Object Selector와 동일한 방법으로 Light 자료를 선택할 수 있도록 하였다.



[그림 3-12] Light Setting 모듈의 기능

나. Light Type Define

광원의 타입은 Directional Light, Positional Light 2 가지 로써 광원에 대한 처리방법에 차이가 있다. Directional Light은 평행광선으로 광원으로부터 물체를 향하는 벡터값은 항상 Vector(0,0,0)으로 벡터값이 있는 Positional Light와는 다르다. 이런이유로 광원 타입이 지정되어야 한다.

다. Ambient Switch Set

한밤중에 방안에 조명을 모두 Off 하여도 희미하게 물체를 감별할 수 있는것과 같은 효과를 주는 기능으로 1 에서 255 까지의 값을 갖고 있다.

라. Light Color Set

광원의 색상은 물체의 색상과 혼합됨으로써 새로운 색상을 만들어 낸다. 광원의 색상도 물체의 색상과 같은 방법으로 인터렉티브하게 R,G,B의 값의 조합을 색상으로 보여주는 방법을 택하였다.

마. Light Location Value Set

광원의 위치(X,Y,Z)를 나타냄으로서 광원이 물체에 영향을 얼마큼 주는가를 계산할 수 있도록 하였다.

바. Light Attribute Switch Set

광원은 Attenuation Light, Spot Light, Cone Light 등의 성질을 달리하는 광원들의 조합으로 나타나게 된다. Attenuate Light = 2, Spot Light = 4, Cone Light = 8 로 값을 줌으로써 Switch의 값에 따라 조합된 조명이 어떤 조명인지를 알 수 있도록 하였다.

사. Spot Light Power Set

Object에 비추는 Highlight에 대한 정도를 표시한다. 1 - 16,383 사이의 값으로 나타내는데 최대로 Spot Light 값은 1 이다.

아. Attenuation Factor Set

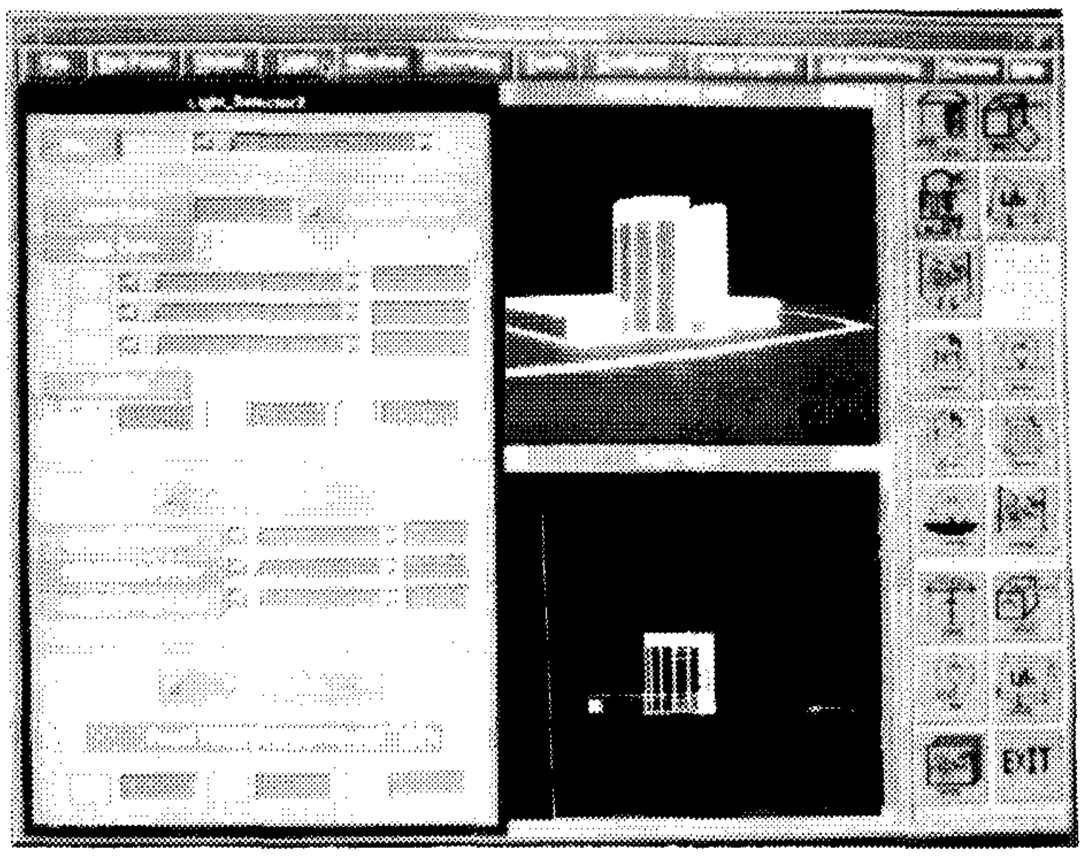
광원과 물체의 거리에 비례하여 점점 어두워지는 비율을 나타내는 계수로 0.0 에서 1.0 까지의 값을 갖는다.

자. Cone Light의 Angle Set

Degree로 표시한다.

차. Target Vector from Light

Positional Light에서만 값을 갖는다. 이 값은 광원이 어느 방향을 비추고 있는가를 나타내는 벡터값(X,Y,Z)으로 표시된다.

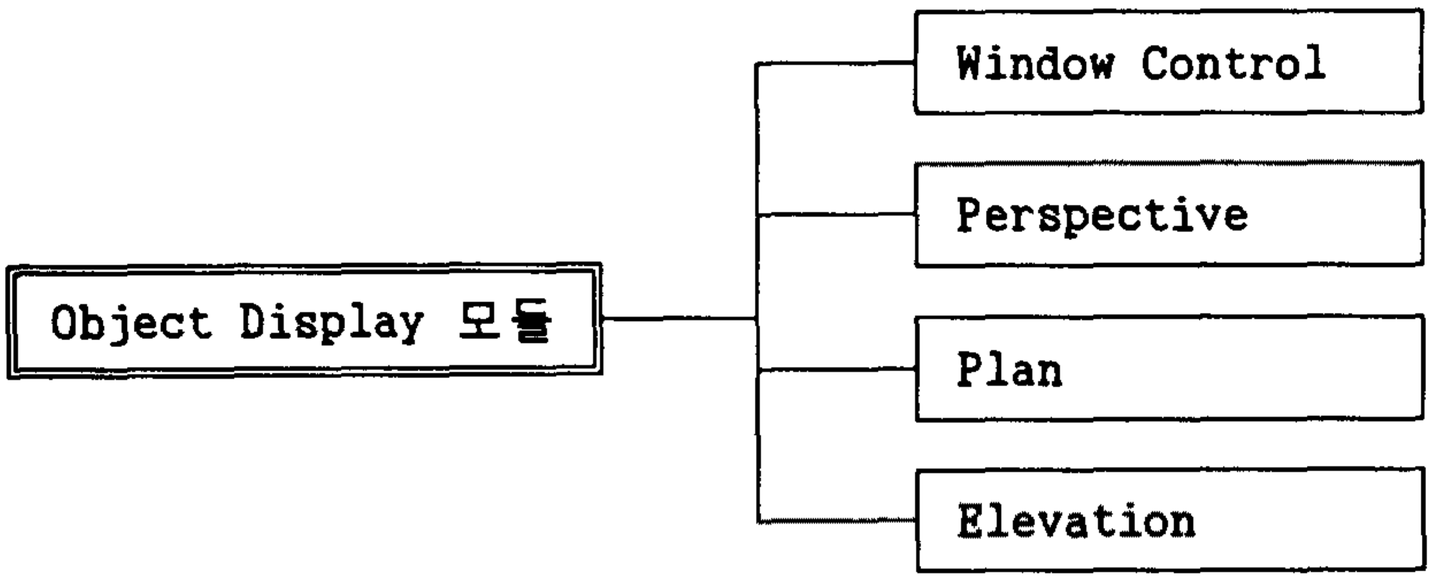


[그림 3-13] Light Setting 모듈의 화면

5. Window(Object Display) 모듈

가. Window Control

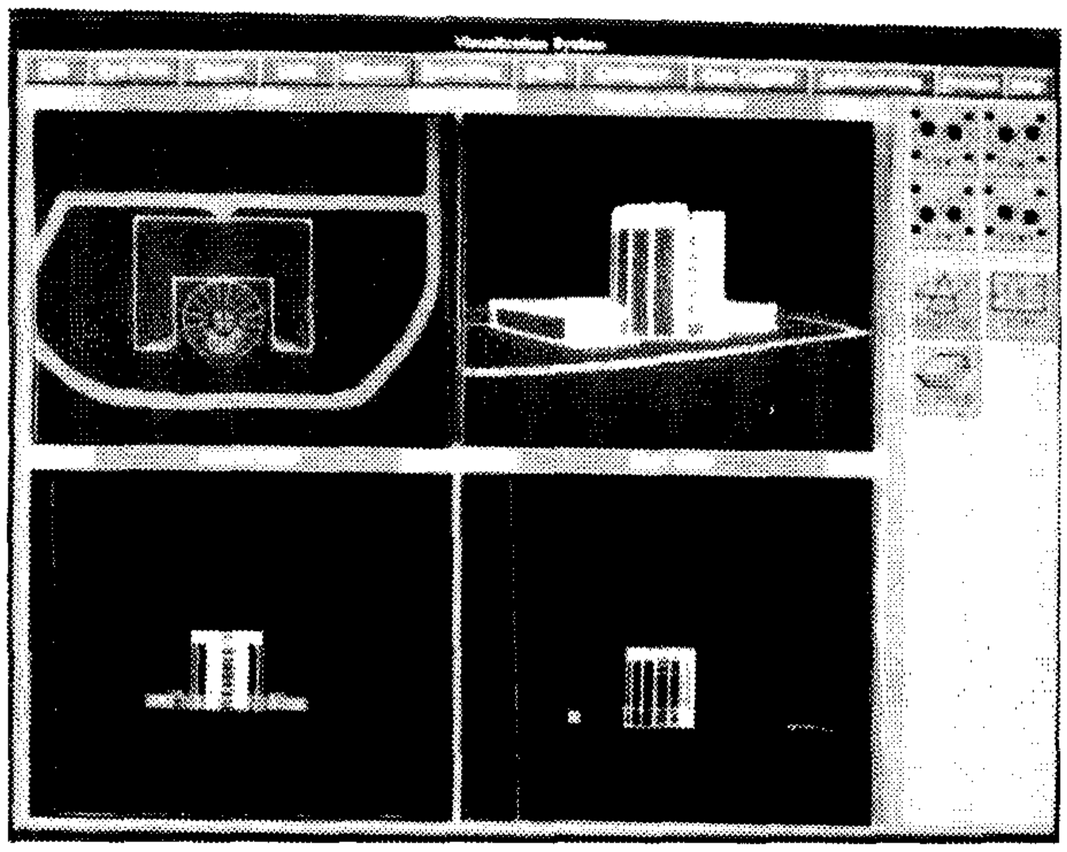
4개의 Window를 만들어 정면도, 입면도, 측면도, 투시도를 함께 볼 수 있도록 한다. 필요에 따라서 원하는 1개의 Window만을 Full Size로 크게 만들어 자세히 볼 수 있도록 하기 위하여 Window라는 Device를 생성, 크기 변경, 위치 변경, 소멸등을 조정하는 하는 기능이다.



[그림 3-14] Window (Object Display) 모듈의 기능

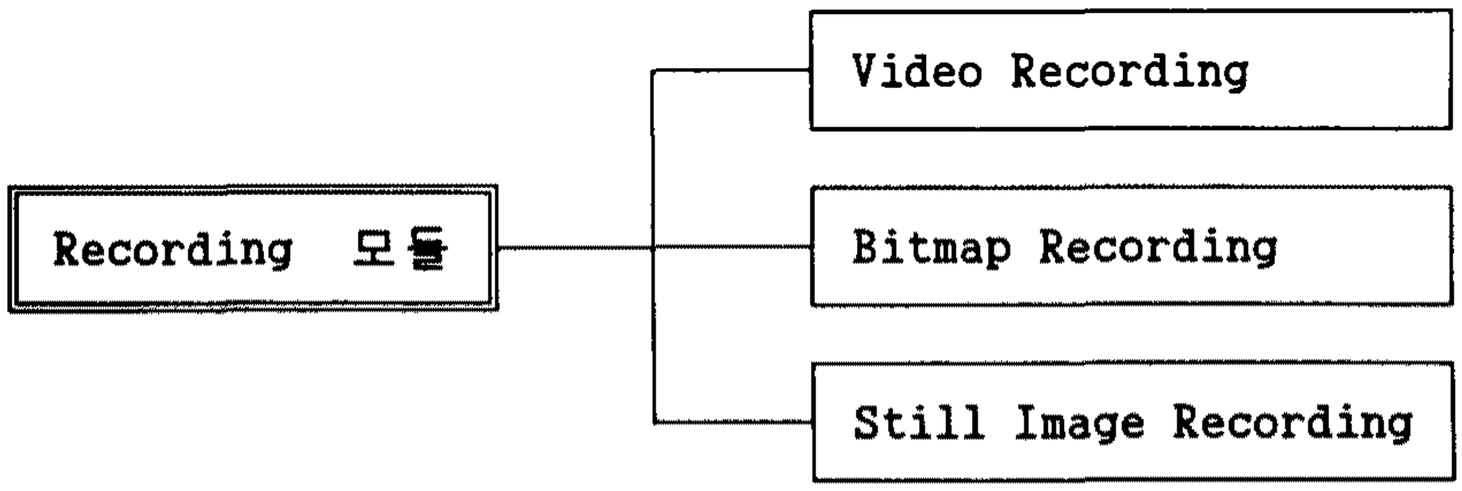
나. Perspective, Plan, Elevation

Perspective는 투시도를 Display 하는 기능이고, Plan은 평면도를 Display 하는 기능을 말하며, Elevation은 입면도 및 측면도등의 단면을 Display 하는 기능을 말한다.

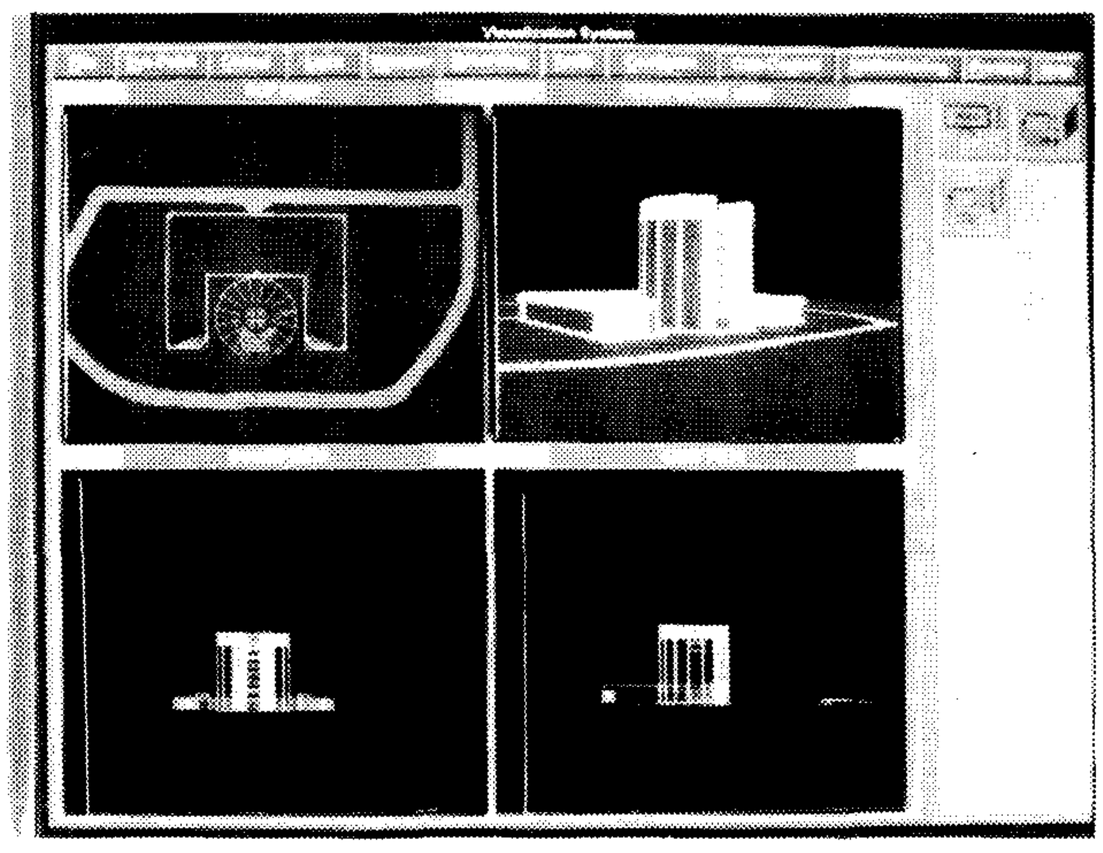


[그림 3-15] Window (Object Display) 모듈의 화면

6. Recording 모듈



[그림 3-16] Recording 모듈의 기능

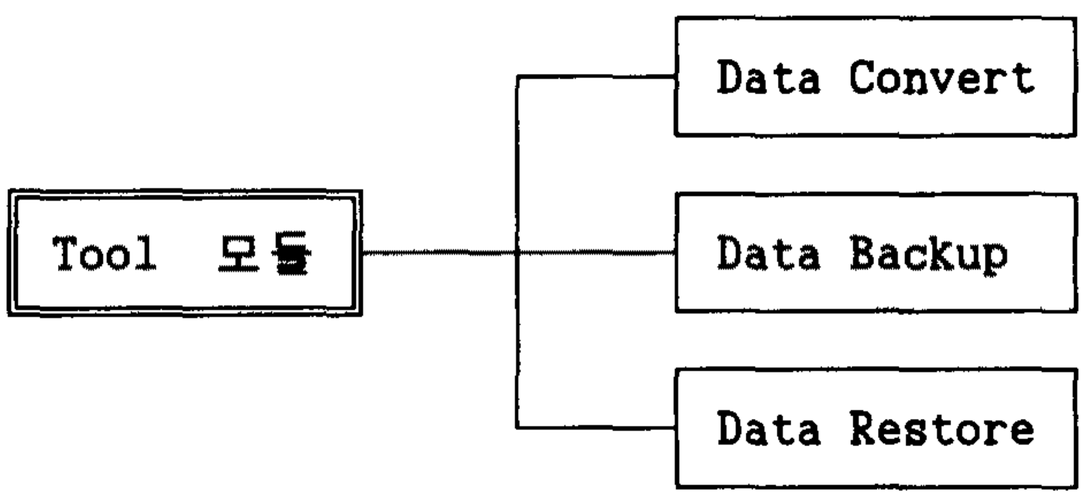


[그림 3-17] Recording 모듈의 화면

Recording 모듈은 2차년도 개발 분야로써 1차년도에는 기본 설계만 하였다. 기능은 RGB 이미지를 Scan Converter등을 통하여 변경된 NTSC를 직접 레코딩하도록 하는 Video Recording 기능, 이미지를 Disk에

Bit Map으로 저장하였다가 필요시 마다 레코딩하는 Bitmap Recording 외에도 카메라로 사진을 찍기 위한 기능인 Still Image Recording 등이 있다.

7. Tool(Utility) 모듈



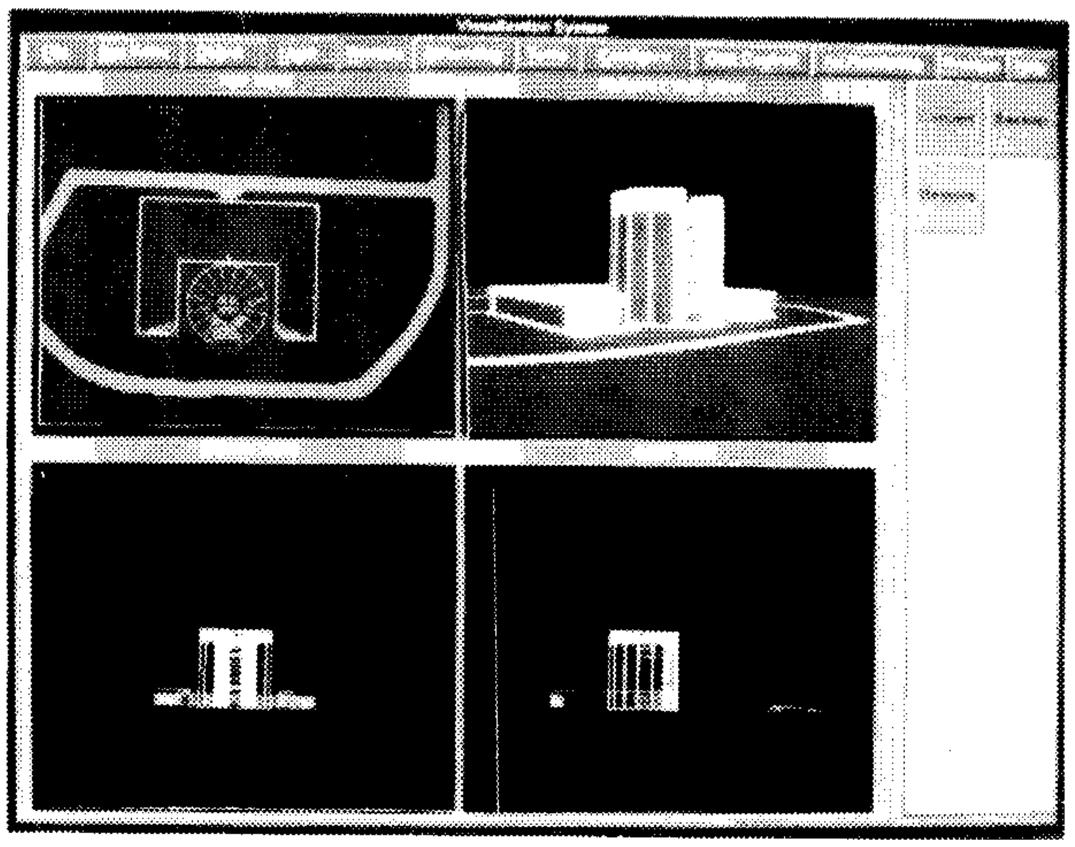
[그림 3-18] Tool 모듈의 기능

가. Data Convert

다른 S/W에서 작업한 자료를 본 시스템에서 이용 가능한 자료로 이용할 수 있도록 자료변환 프로그램을 개발하여 사용한다.

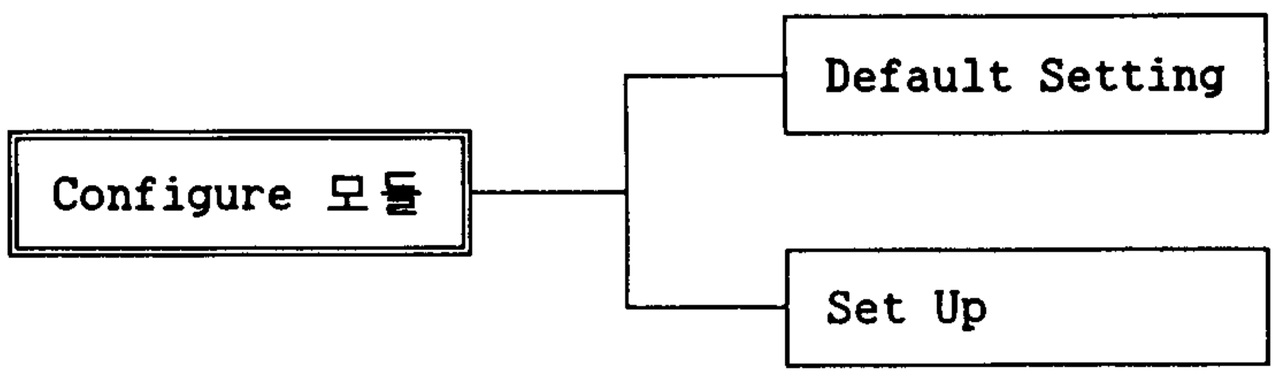
나. Data Backup 및 Restore

방대한 이미지 Data를 Internal Disk 에서 External Disk로 보내는 Backup 기능과 그 반대의 기능인 Restore 기능을 말한다.



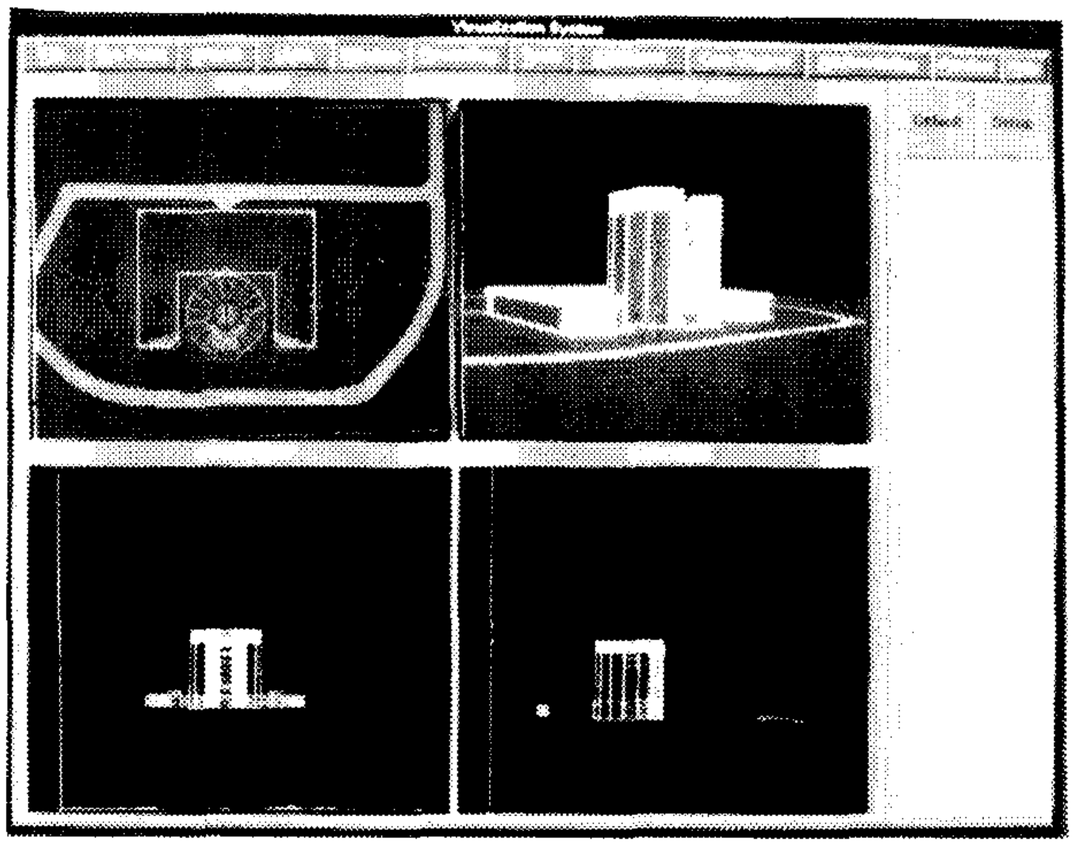
[그림 3-19] Tool 모듈의 화면

8. Configure 모듈



[그림 3-20] Configure 모듈의 기능

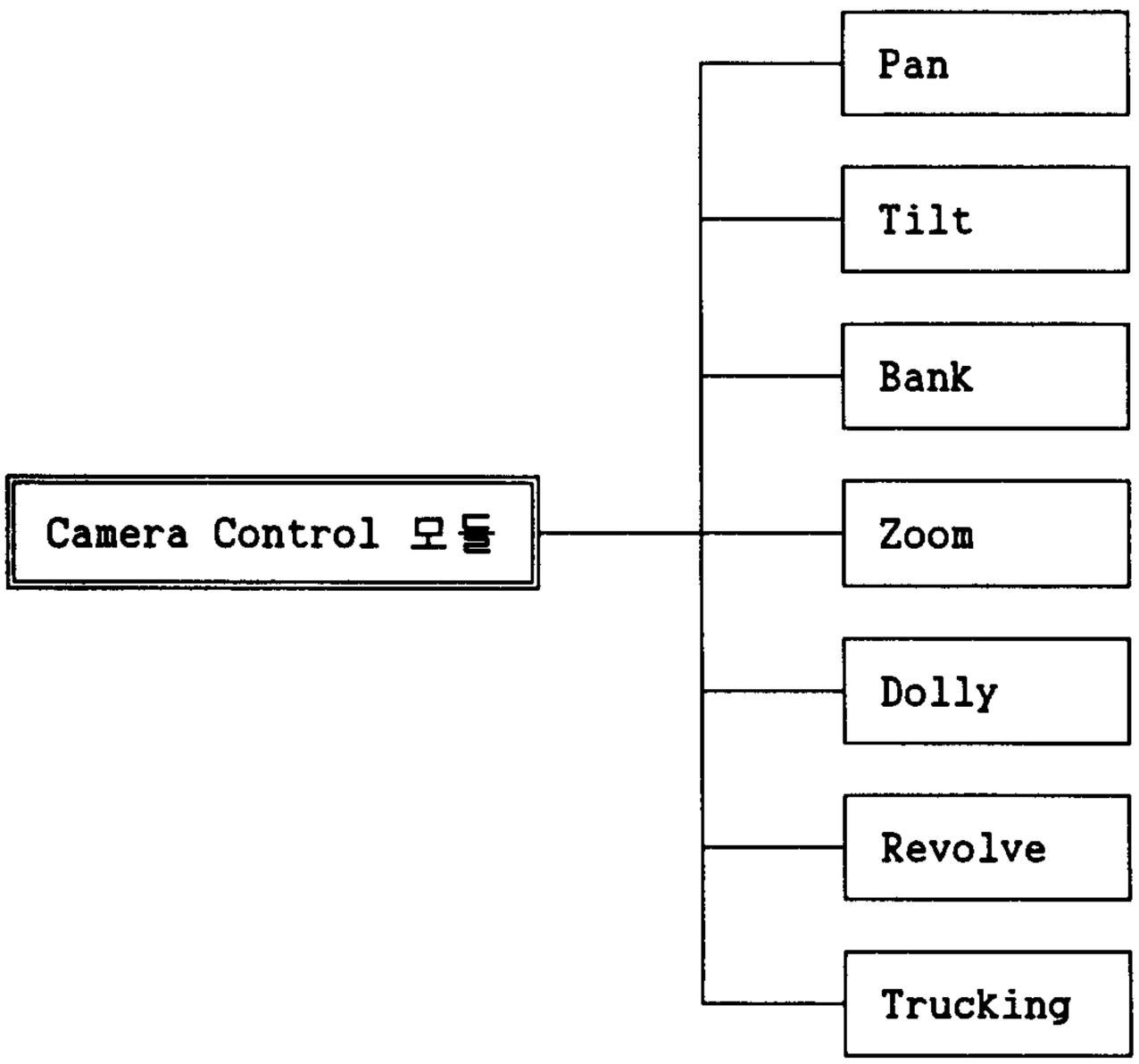
Configuration 모듈은 Device Setting을 위한 모듈로써 기본적으로 프로그램 수행에 필요한 Default Setting 기능과 필요시마다 Default 값을 변경하여 사용하는 Set Up 기능이 있다.



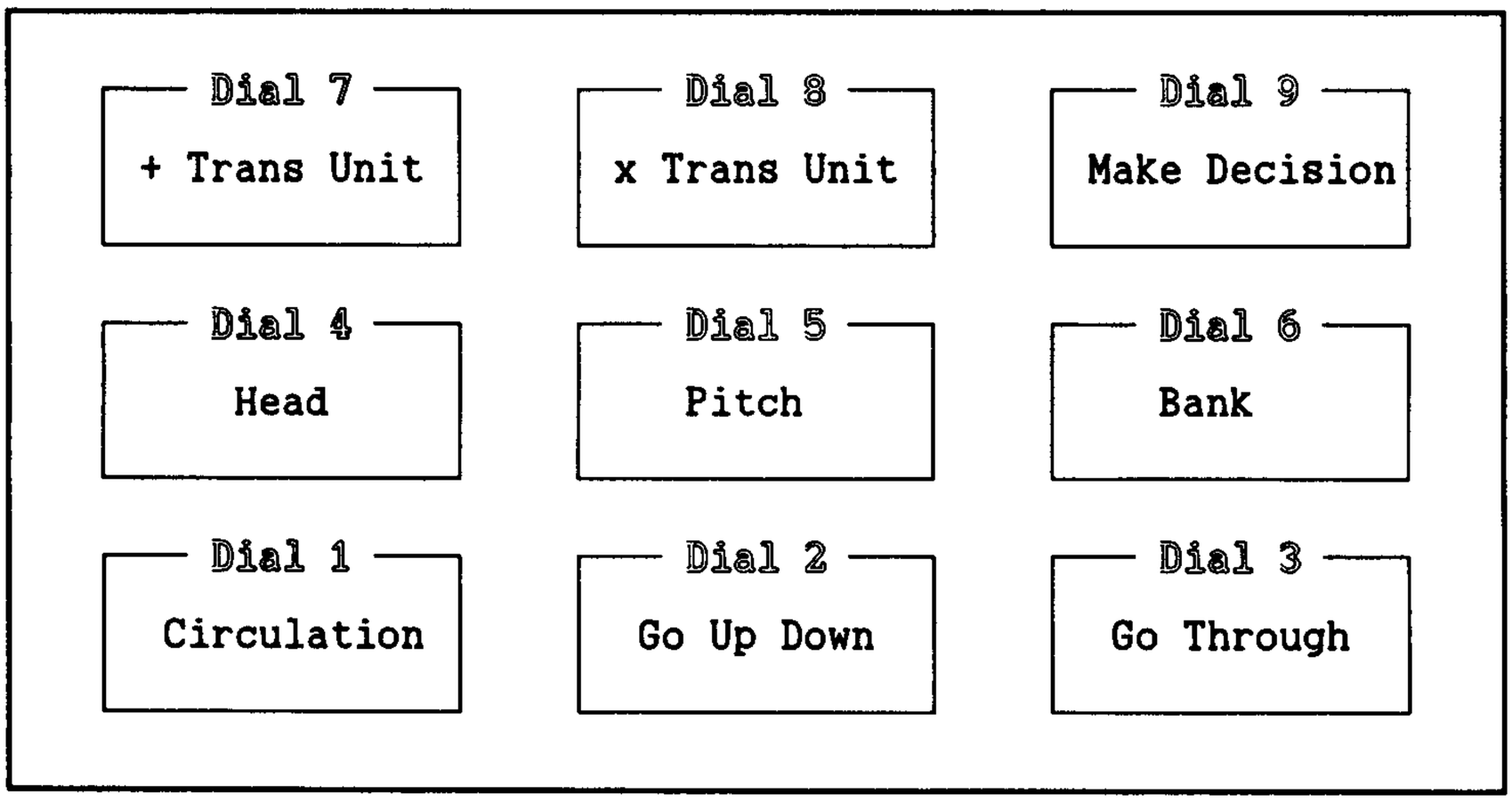
[그림 3-21] Configure 모듈의 화면

9. Camera Control 모듈

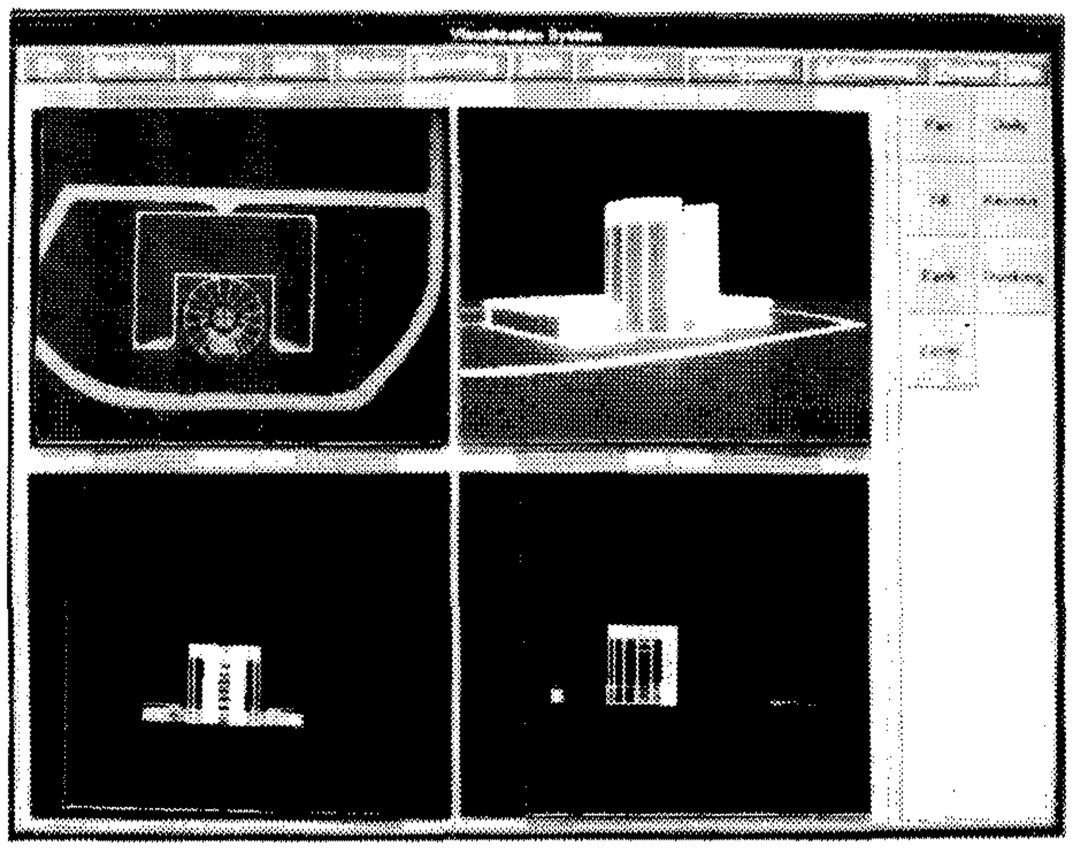
카메라의 움직임은 촬영기법의 기본동작인 Pan, Tilt, Bank, Zoom, Dolly, Revolve, Trucking등을 메뉴로 나타내고 마우스로 Picking하여 카메라의 이동을 마우스로 처리하는 방법과 9개의 다이알 박스에 각각의 기능을 정의하여 카메라의 이동을 처리하는 방법 중 사용자가 원하는 방법을 선택하여 사용할 수 있도록 하였다.



[그림 3-22] Camera Control 모듈의 기능



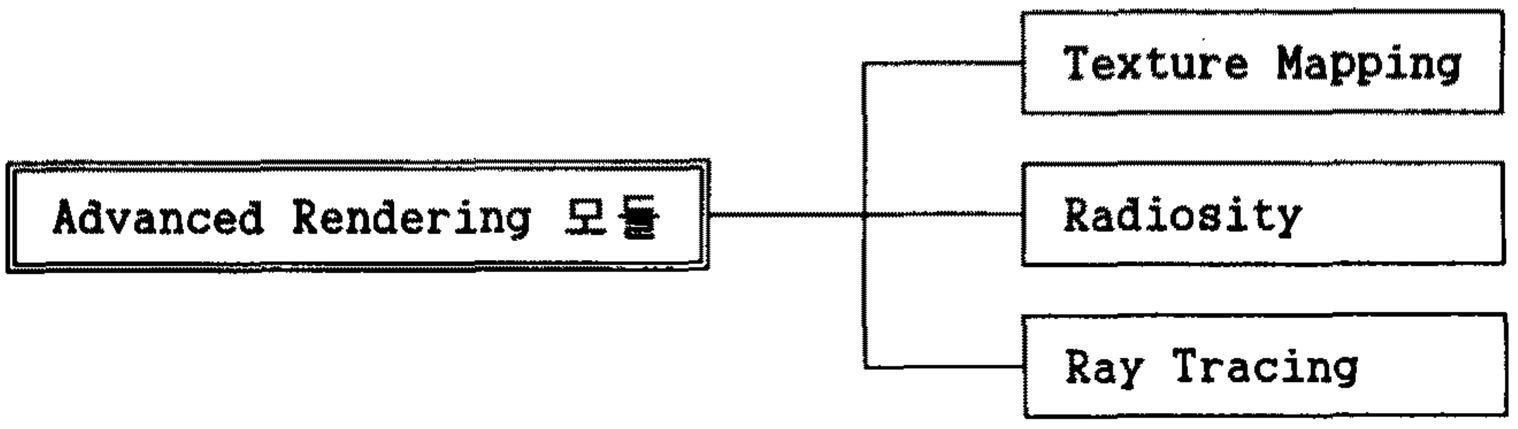
[그림 3-23] 9개 Dial에 카메라 움직임 정의



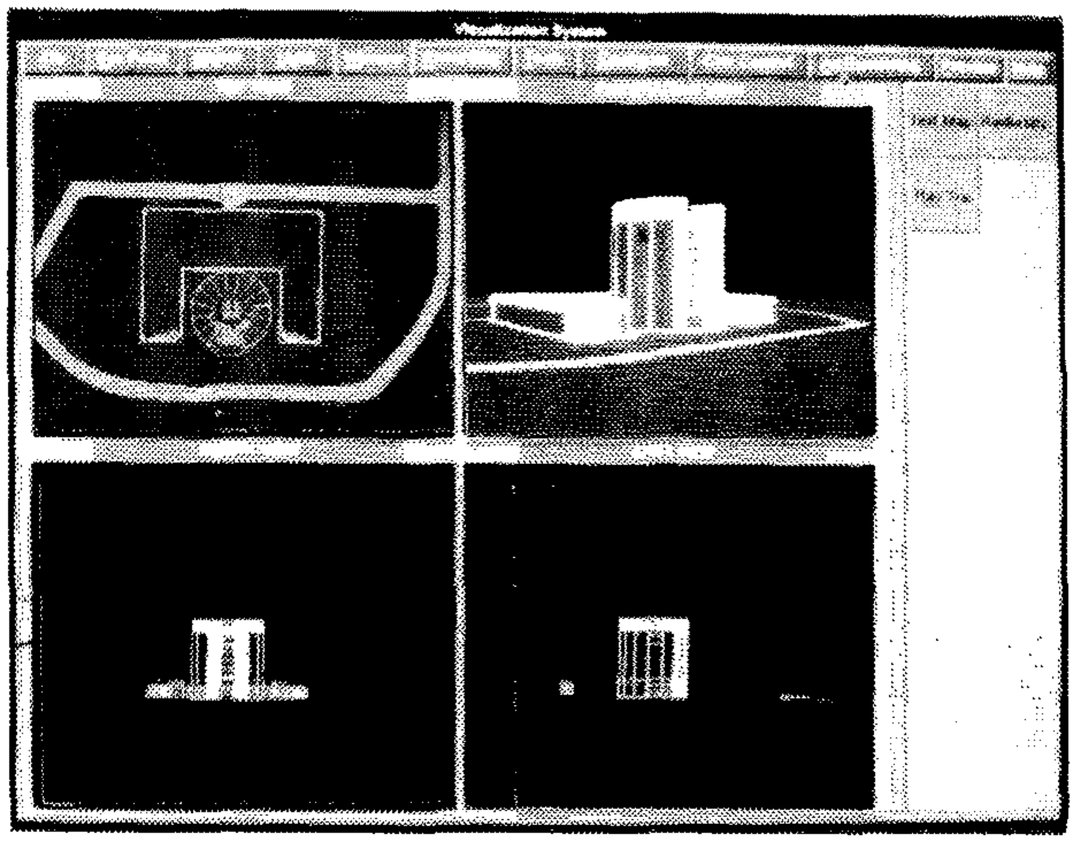
[그림 3-24] Camera Control 모듈의 화면

10. Advanced Rendering 모듈

2차년도 연구 분야로써 고도의 렌더링 알고리즘에 의해 프로그램을 개발한다.



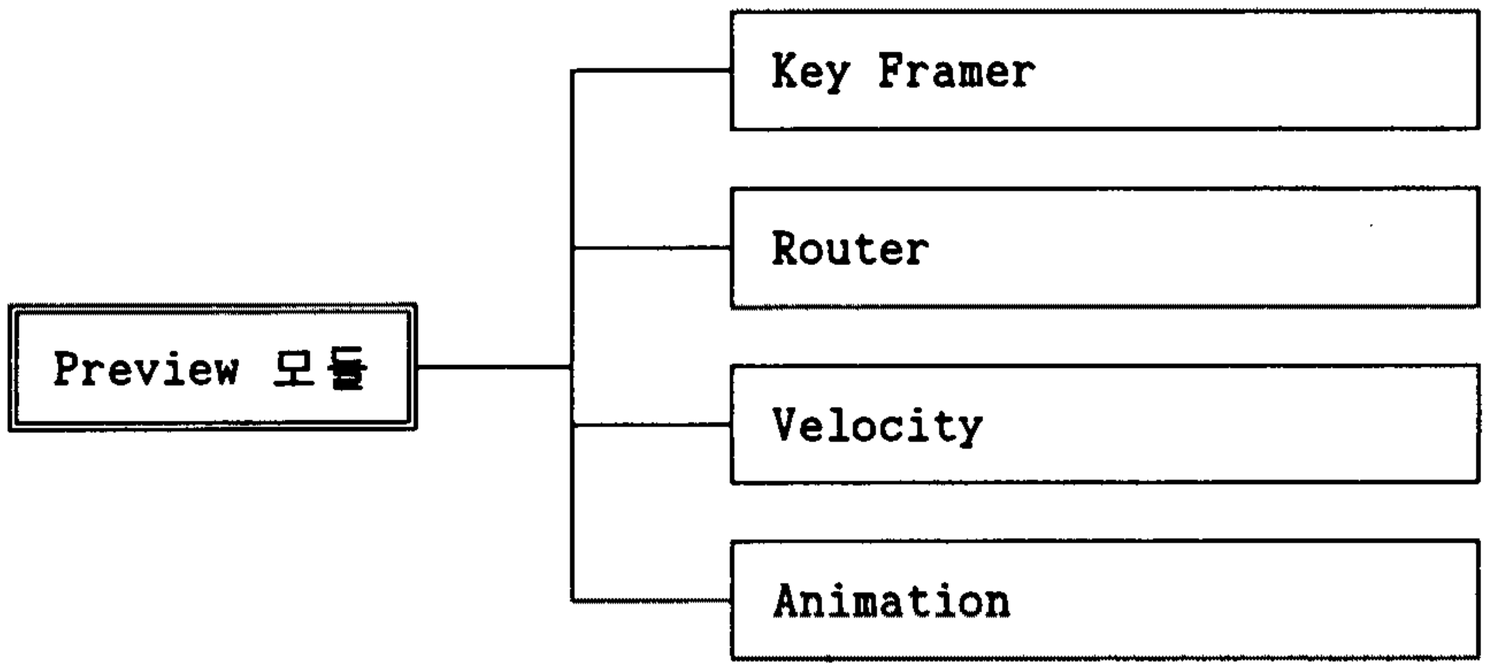
[그림 3-25] Advanced Rendering 모듈의 기능



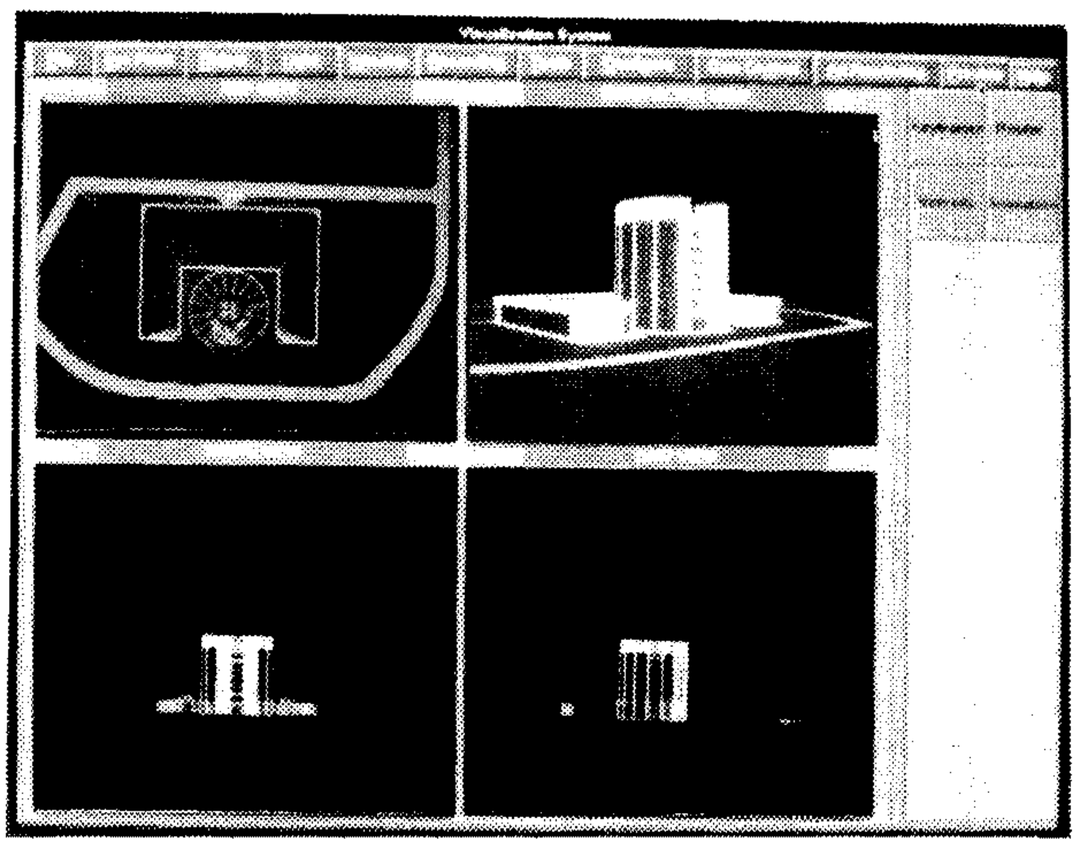
[그림 3-26] Advanced Rendering 모듈의 화면

11. Preview 모듈

2차년도 연구분야로 Key Framer, Router, Velocity, Animation 기능을 단계적으로 연구한다.



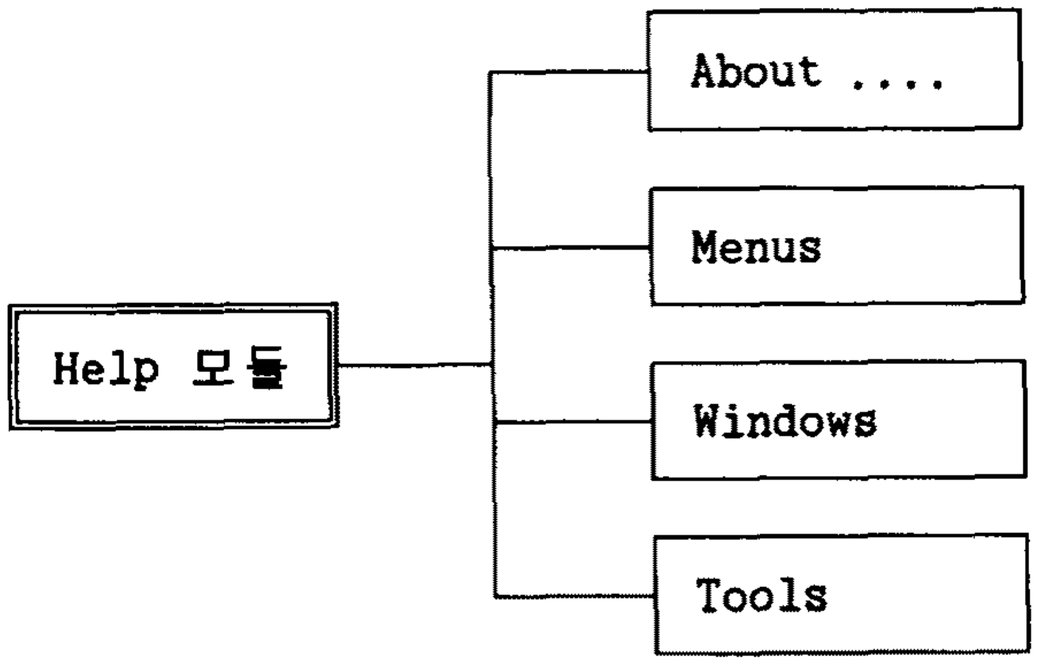
[그림 3-27] Preview 모듈의 기능



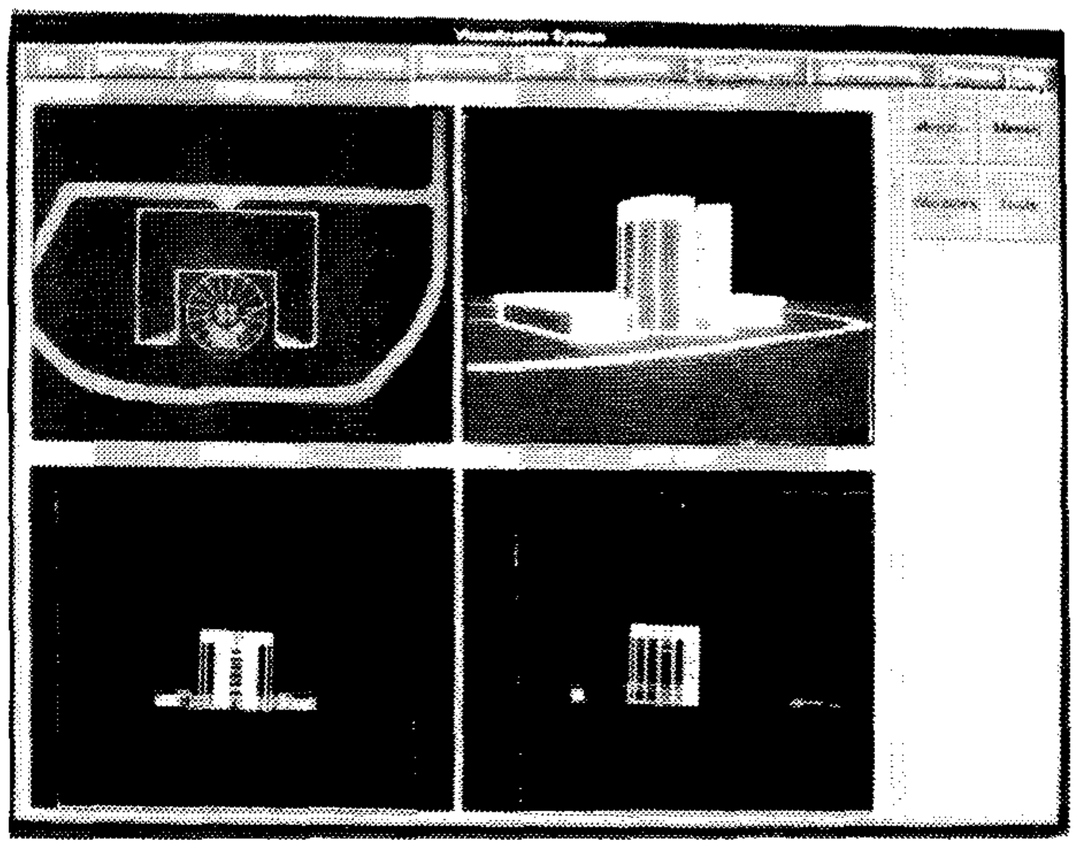
[그림 3-28] Preview 모듈의 화면

12. Help 모듈

Manual 없이도 조작이 가능하도록 도움말 모듈을 개발하였다



[그림 3-29] Help 모듈의 기능



[그림 3-30] Help 모듈의 화면

제 5 절 OSF/Motif로 구현한 화면의 계층적 구조

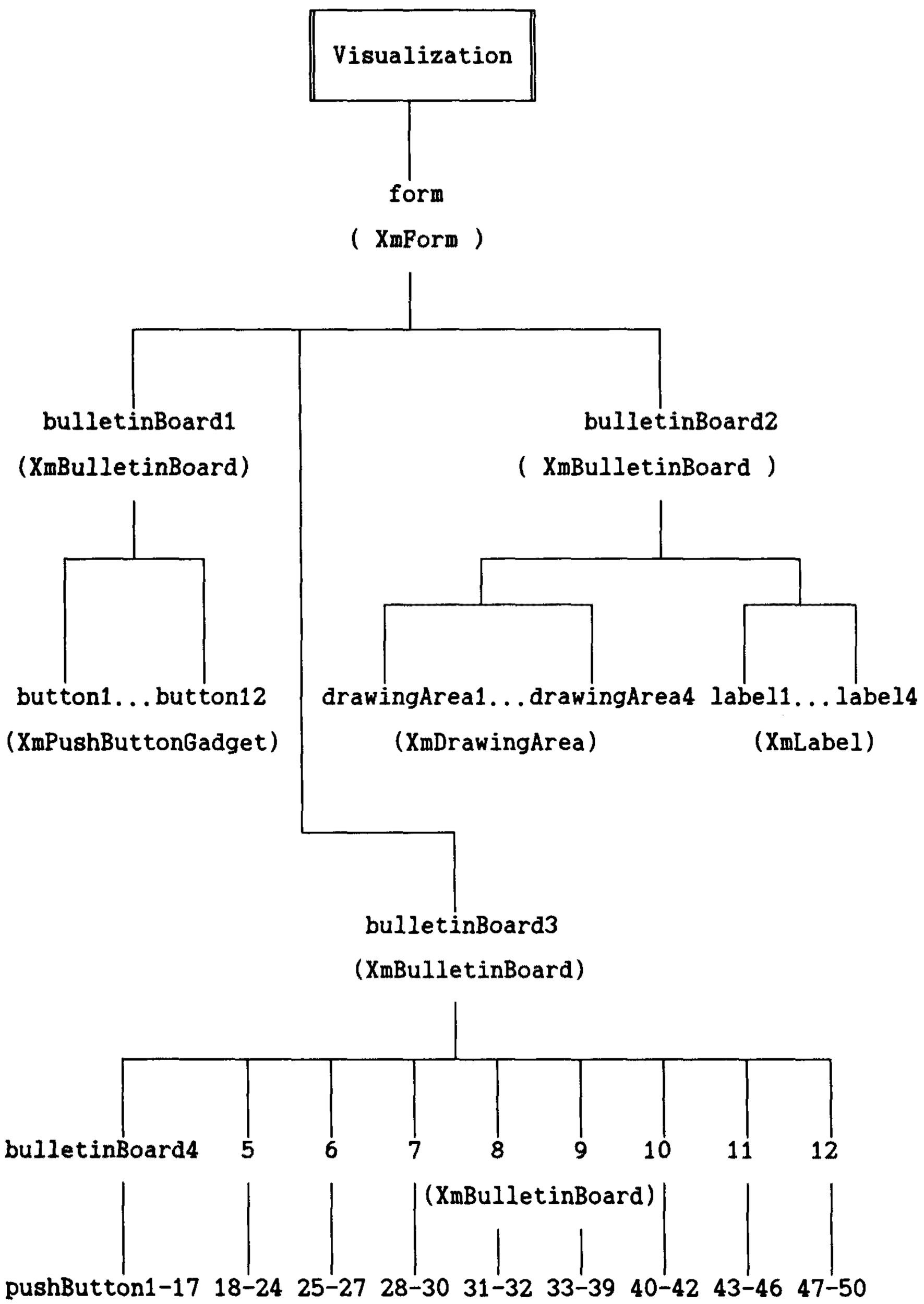
본 연구에서 활용한 Graphics User Interface Tool인 OSF/Motif는 워크스테이션 분야에서의 X-window프로그램 개발을 위한 표준 tool로 정착되어 가고 있기 때문에 본 연구에서는 OSF/Motif를 사용하였다.

다음은 Visualization S/W를 구성하는 Widget들의 계층적 구조를 나타낸것이다.

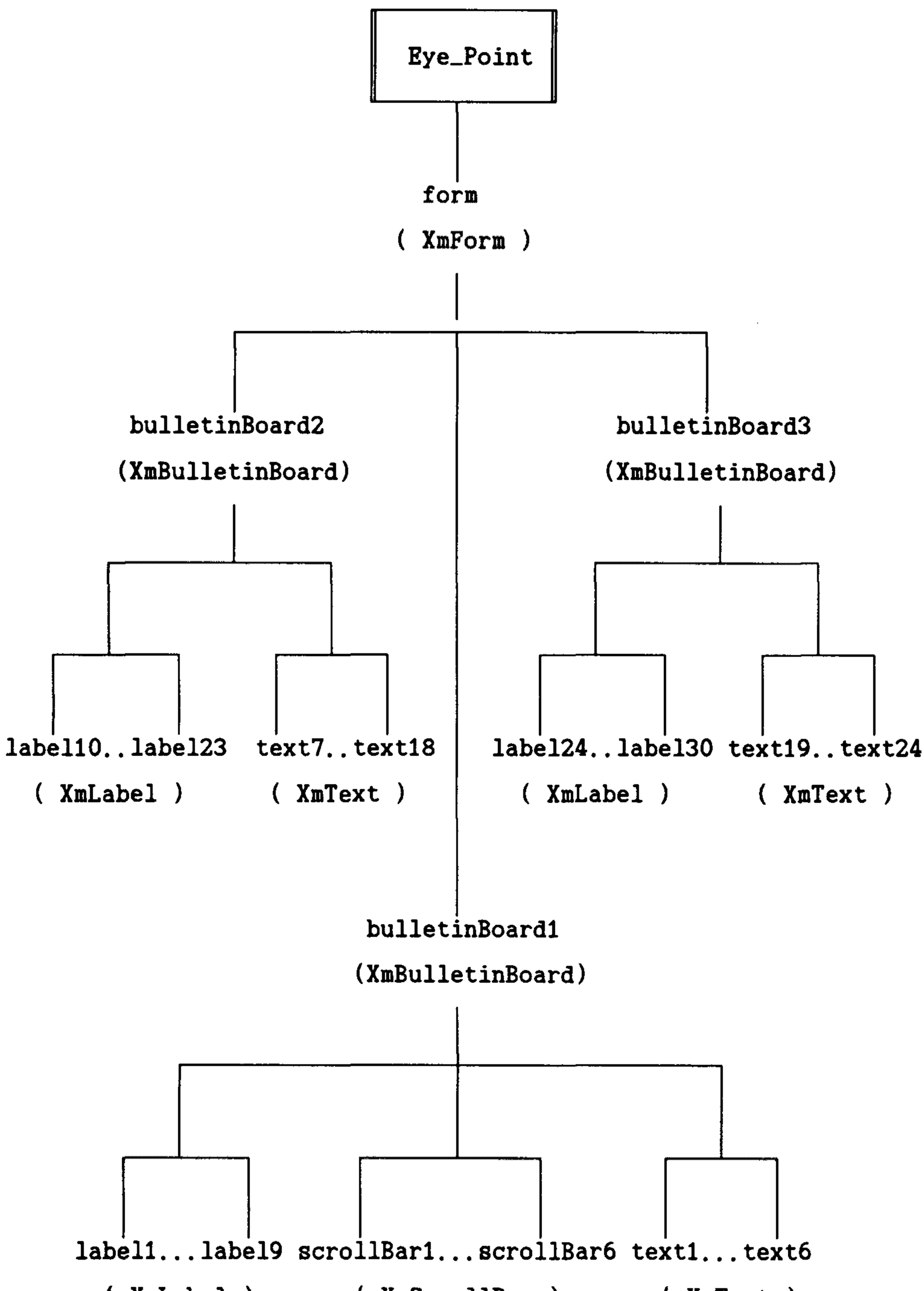
메인 윈도우의 Widget 들의 계층적 구조는 [그림 3-31]와 같이 Form Widget 위에 3개의 BulletinBoard Widget으로 구성되며, 각각의 Bulletin Board Widget 위에는 PushButton Gadget과 DrawingArea Widget 등의 여러 가지 Widget들이 조합되어 구성되어있다. Top View, East View, South View, Perspective View 등의 View를 나타낼 수 있는 4개의 DrawingArea를 갖고있다.

Eye_Point의 Widget에 대한 계층도는 [그림 3-32]와 같다. 3개의 BulletinBoard Widget으로 구성되며, 각각의 BulletinBoard Widget은 Label과 Text 및 ScrollBar로 이루워져있다.

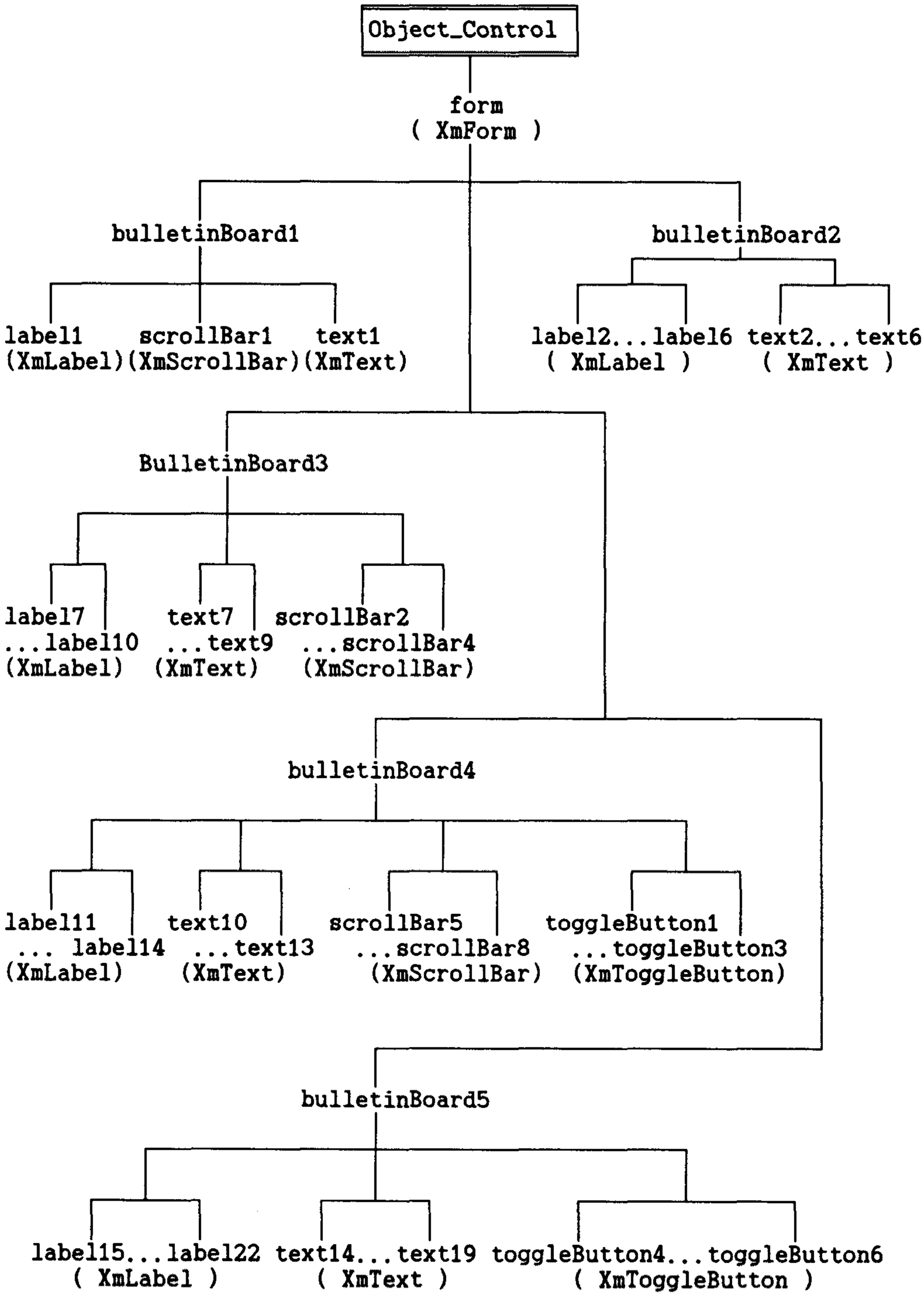
Object_Control에 대한 계층도는 [그림 3-33] 이고, Light Setting의 Widget들에 대한 계층적구조는 [그림 3-34] 이다.



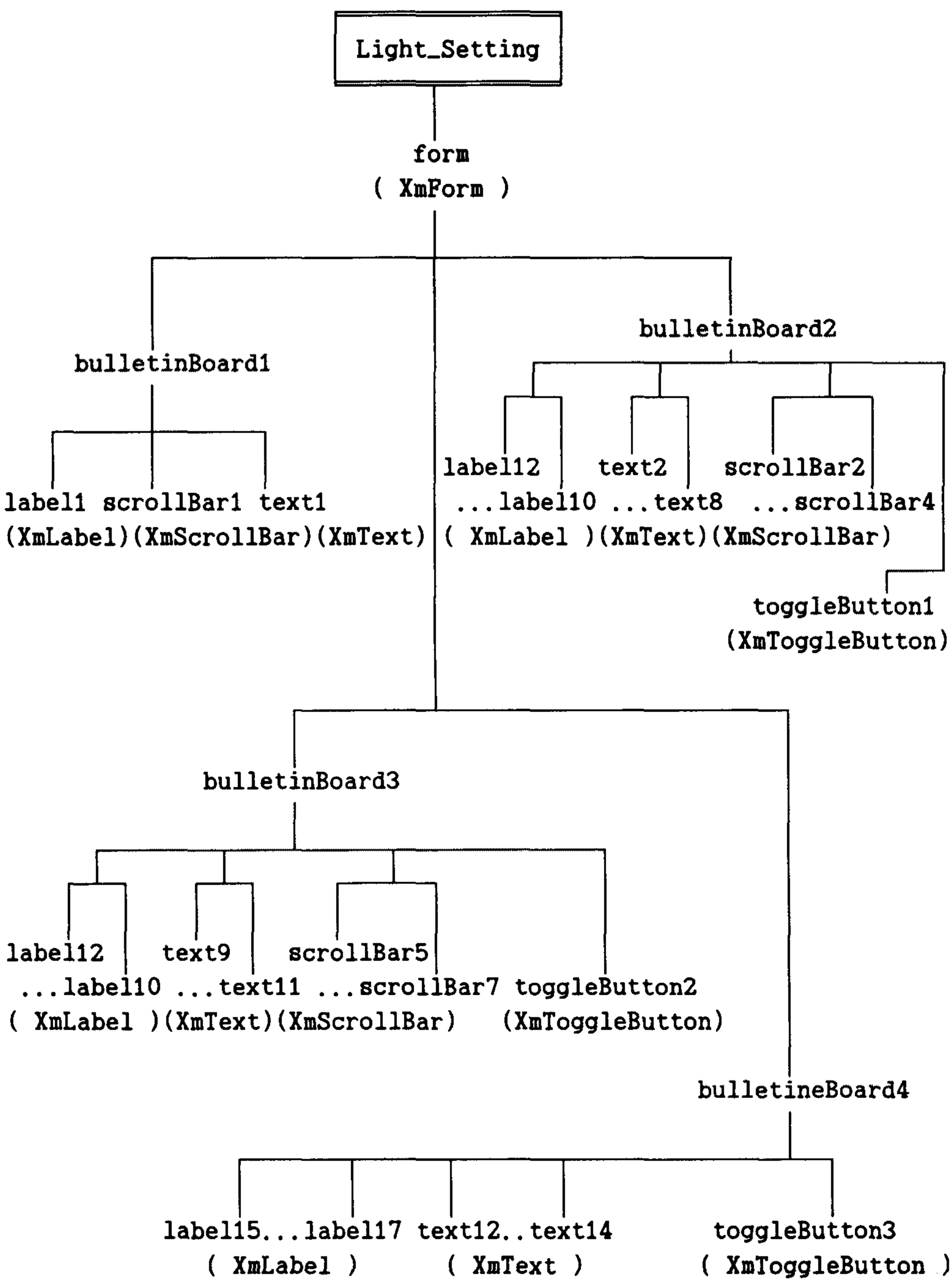
[그림 3-31] Main Menu 에서의 Widget들의 계층적 구조



[그림 3-32] Eye_Point 메뉴에서의 Widget들의 계층적 구조



[그림 4-33] Object_Control 메뉴에서의 Widget들의 계층적 구조

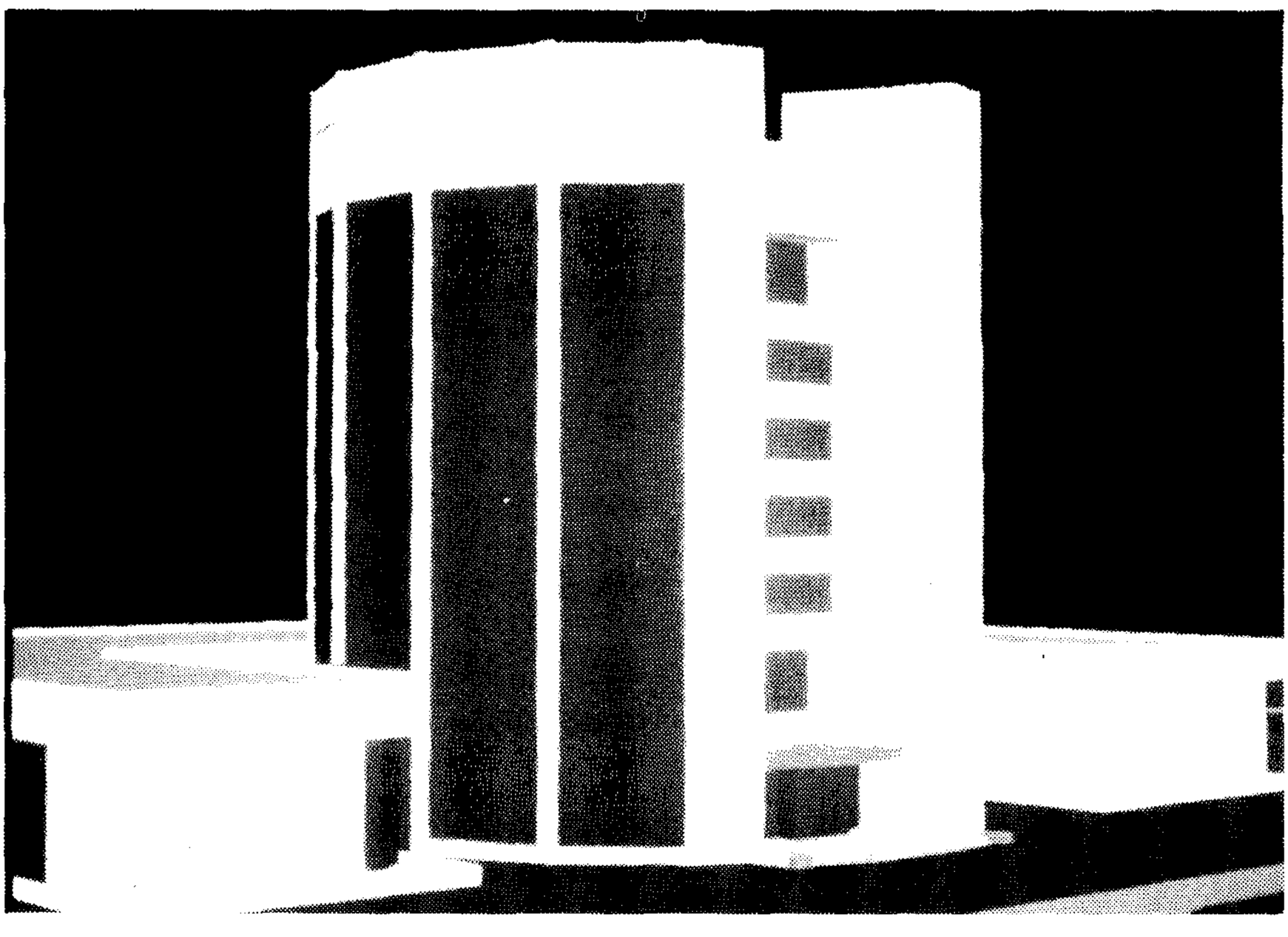


[그림 3-34] Light_Setting 메뉴에서의 Widget들의 계층적 구조

제 4 장 개발된 시스템의 실무적용

제 1 절 시스템공학 연구소 건물의 3차원 표현

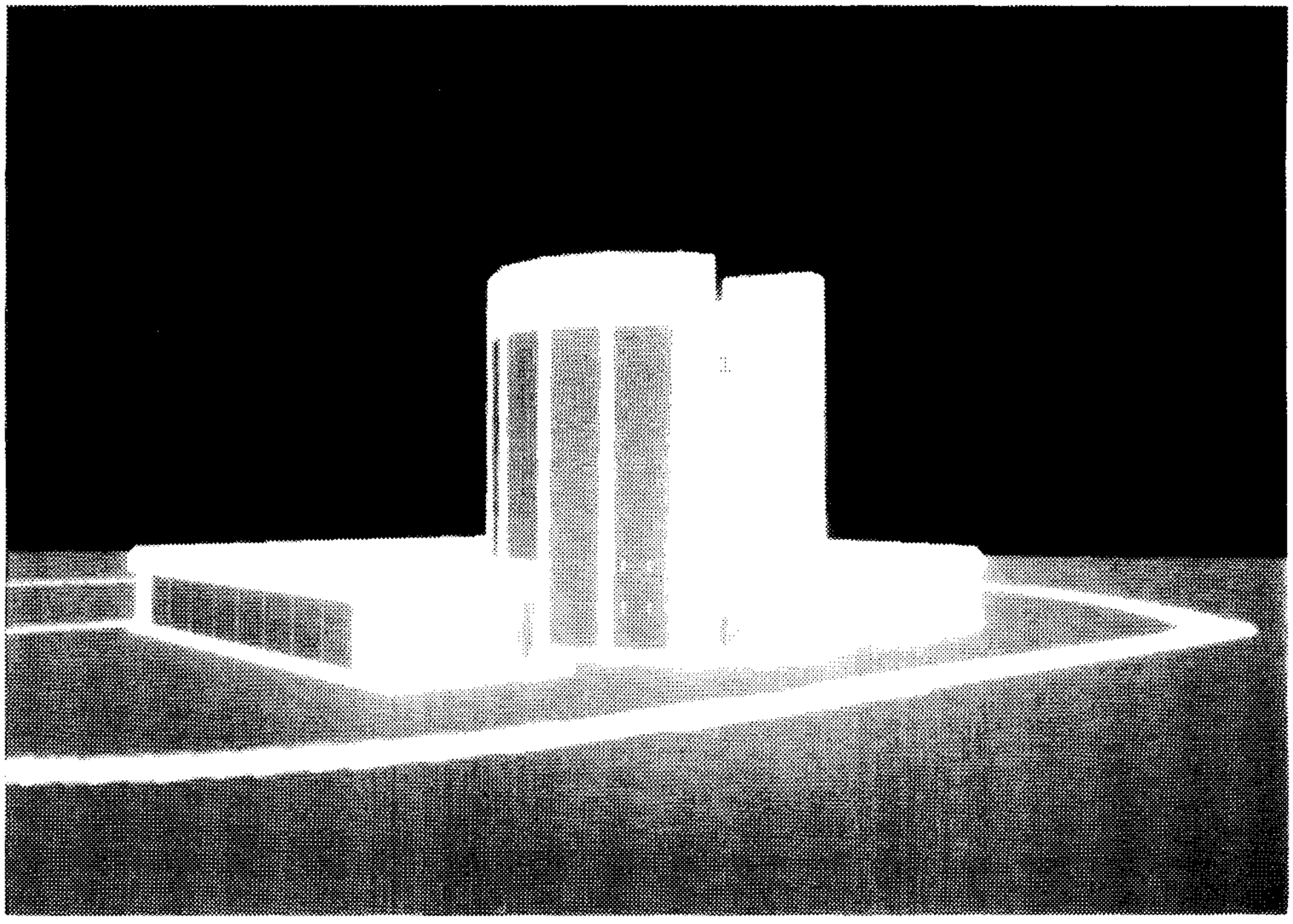
본 Visualization 시스템의 Viewing 및 Coloring 모듈 Test와 시스템의 문제점을 파악하기 위하여 당 연구소의 건물을 3차원으로 표현하는 작업을 실시하였다. 시스템공학 연구소 건물은 16각형 건물로써 일반적인 직육면체 건물에 비하여 모델링작업에 소요되는 작업시간이 많았다.



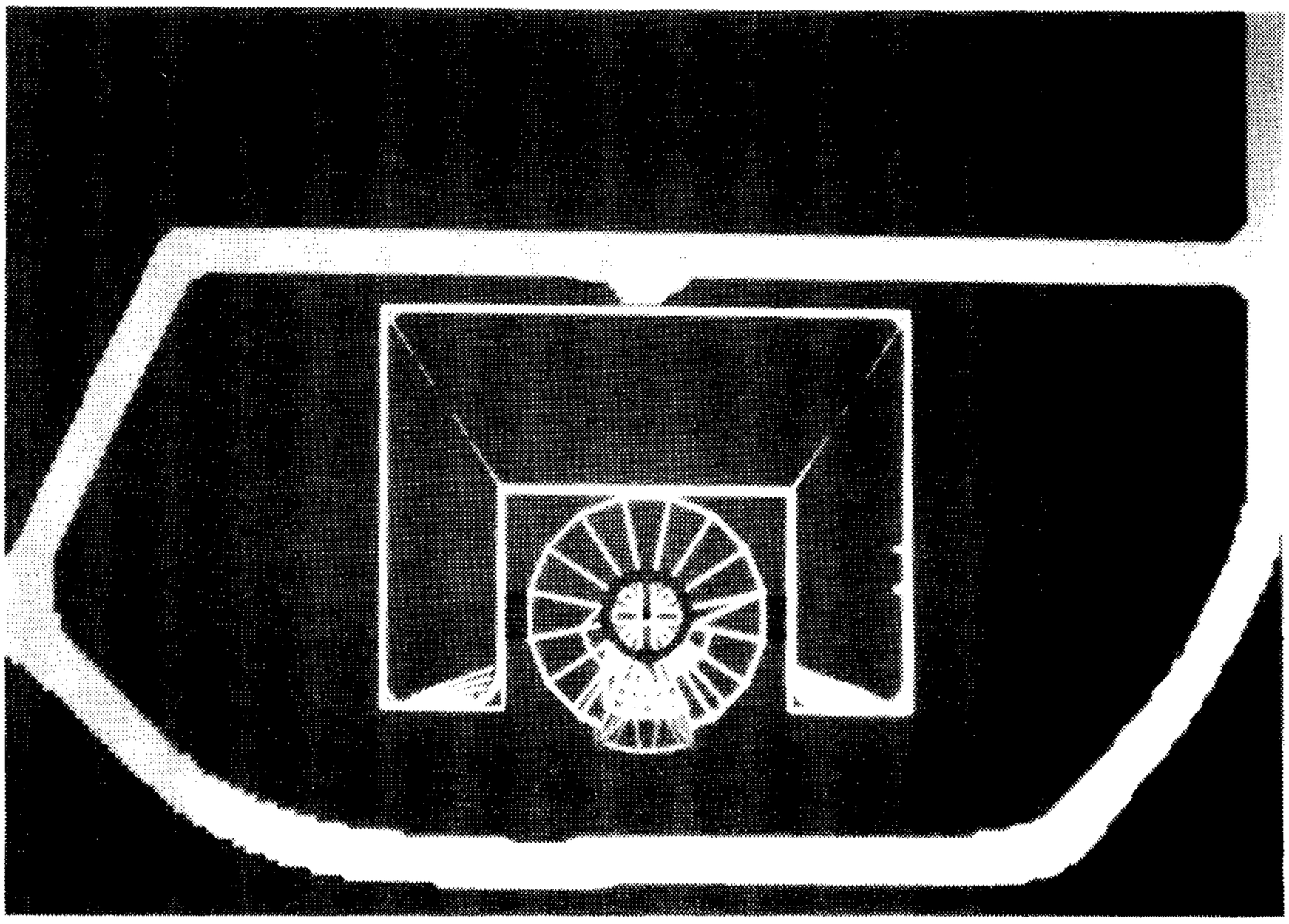
[그림 4-1] 시스템공학 연구소 전경(I)

본 작업의 목적은 S/W의 TEST 외에도 당 연구소의 정문을 만들기 전에 정문과 건물과의 조화를 시뮬레이션하기 위함이였다. 그러나, 본 정문을 만들기 위한 작업이 보류되고, 본 연구의 1차년도 연구종료일이 다가옴에 따라 당 연구소 건물만을 3차원으로 표현하였다.

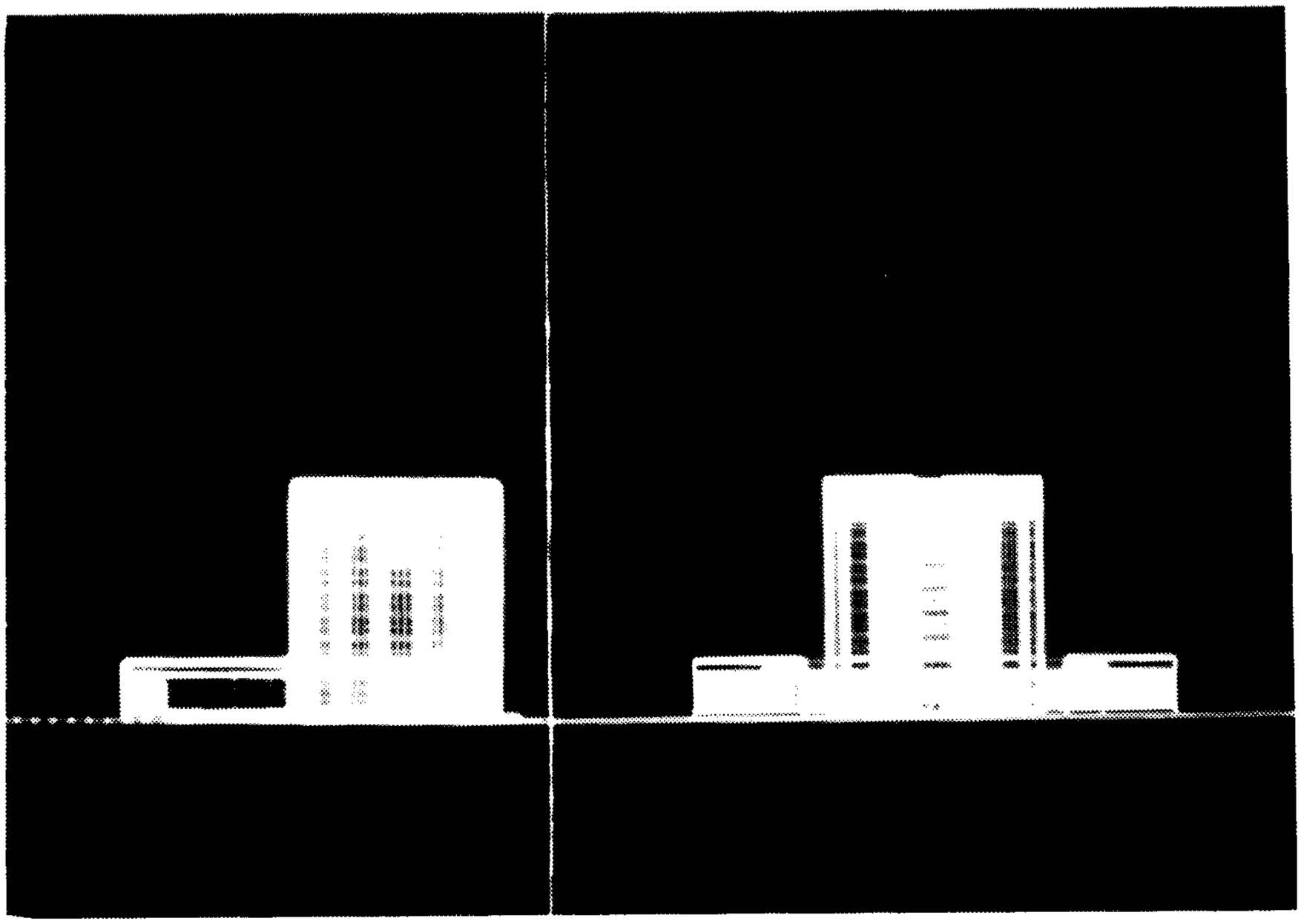
[그림 4-1] 화면을 Focus Value 조정 모듈을 거친 화면 [그림4-2]을 Display 하여 보았고, 이 화면에 대한 정면도,입면도,측면도를 나타내보았다. 즉, [그림 4-3]는 위에서 내려다 본 Top View 이고, [그림 4-4]는 측면에서 바라본 Top View 와 정면 바라본 View 이다.



[그림 4-2] 시스템공학 연구소 전경(II)



[그림 4-3] 시스템공학 연구소 입면도



[그림 4-4] 시스템공학 연구소 정면도와 측면도

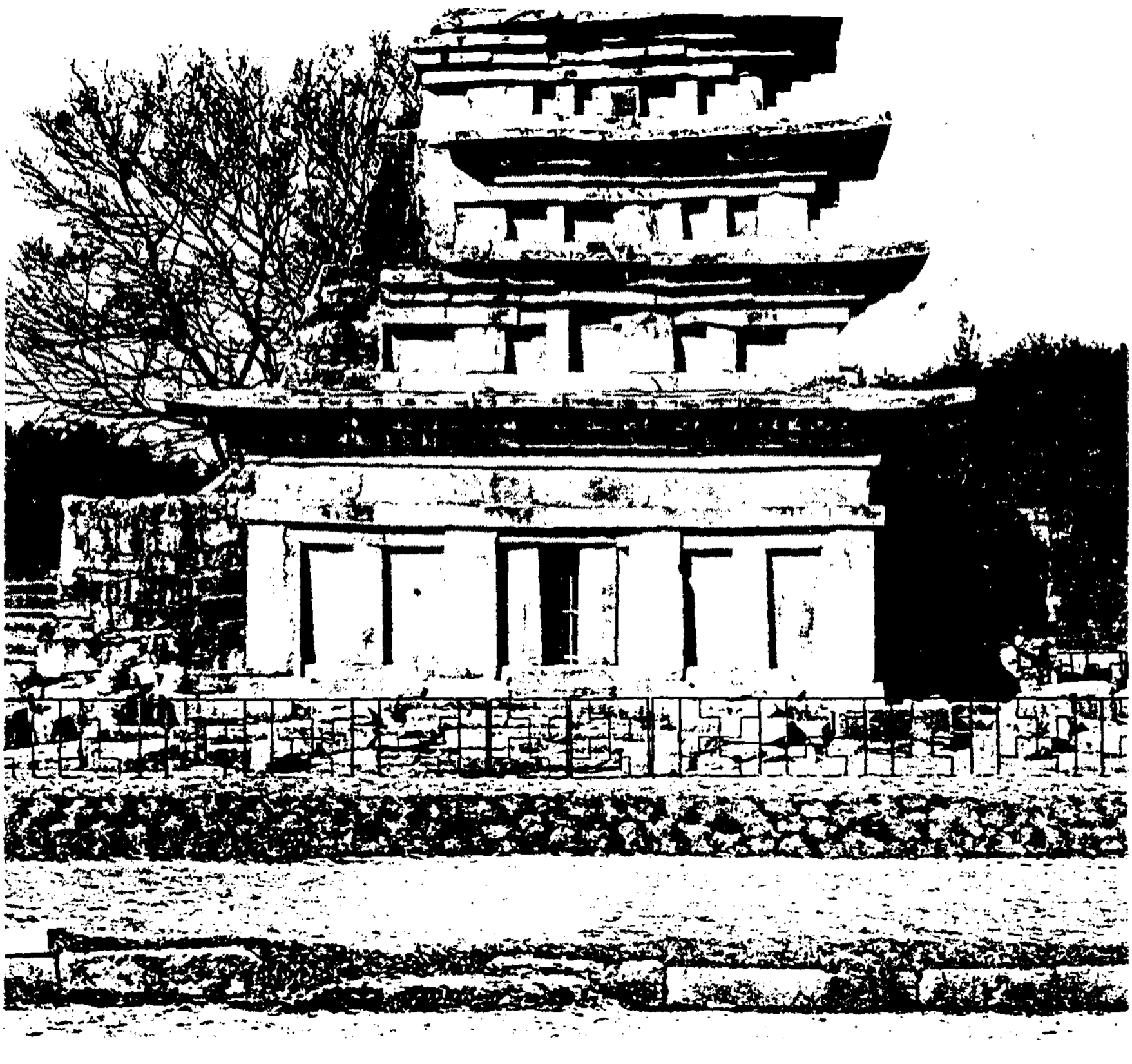
제 2 절 3차원 컴퓨터그래픽을 이용한 미륵사지 서탑복원

『Visualization S/W 개발에 관한 연구』와 『컴퓨터그래픽 기술을 이용한 미륵사지 서탑복원에 관한 연구』가 병행하여 이루워짐으로 본 시스템의 개발에 크게 2가지의 도움이 있었다.

첫번째는 본 연구의 목표인 세계적으로 유명한 Visualization S/W의 국산화를 실현하기 위하여 Wavefront라는 과학및 공학분야로 가장 널리 알려진 Visualization S/W 를 사용 할 기회가 생김으로 인하여 본 S/W의 개발 방향을 확실히 할 수 있었으며, 두번째는 실제의 연구에 본 S/W를 사용해 볼 수 있는 기회가 주워짐으로써 본 S/W를 여러면에서 테스트 해 볼 수 있었다는 점이다.

전북 익산에 위치한 미륵사지 서탑은 국보 제 11호로 지정된 백제때의 건축된 탑으로 [그림 4-5]와 같이 6층의 일부분만 남아 흉칙한 모습을 하고 있다. 일제는 파손된 미륵사지 서탑을 시멘트로 대강 보강하여 놓았다. 이는 우리의 전통문화유산 보잘것 없다는것을 간접적으로 보여주는 효과를 갖는 일제의 문화말살정책의 일환이였다는 주장이 지배적이다.

이와같은 주장으로 인하여 미륵사지 서탑의 복원을 위한 계획이 발표되었으나, 우선 3차원 컴퓨터그래픽으로 복원 후의 모습을 표현하는 작업이 선행되어야 한다는 문화부의 요청에 따라 연구가 진행되었다.



[그림 4-5]　미륵사지 서탑의 현재 모습

컴퓨터그래픽 애니메이션

컴퓨터 시뮬레이션 기술

전통문화유산

연구내용

o 전통문화유산과 3차원 컴퓨터그래픽 애니메이션 기술을 접목
o 복원 前에 복원도에 의하여 복원 後의 모습을 컴퓨터에 표현
o 시공과정 시뮬레이션 -> 복원도의 오류검색과 고증작업 자료

기대효과

o 시행착오 배제 :
복원시 시행착오를 줄여 인건비, 작업시간 절감
o 문화재 홍보 :
첨단기술의 접목으로 문화재에 대한 새로운 관심
o 문화재 보존 :
3차원 컴퓨터그래픽기술에 의한 자료의 영구보존

[그림 4-6] 연구내용 및 기대효과

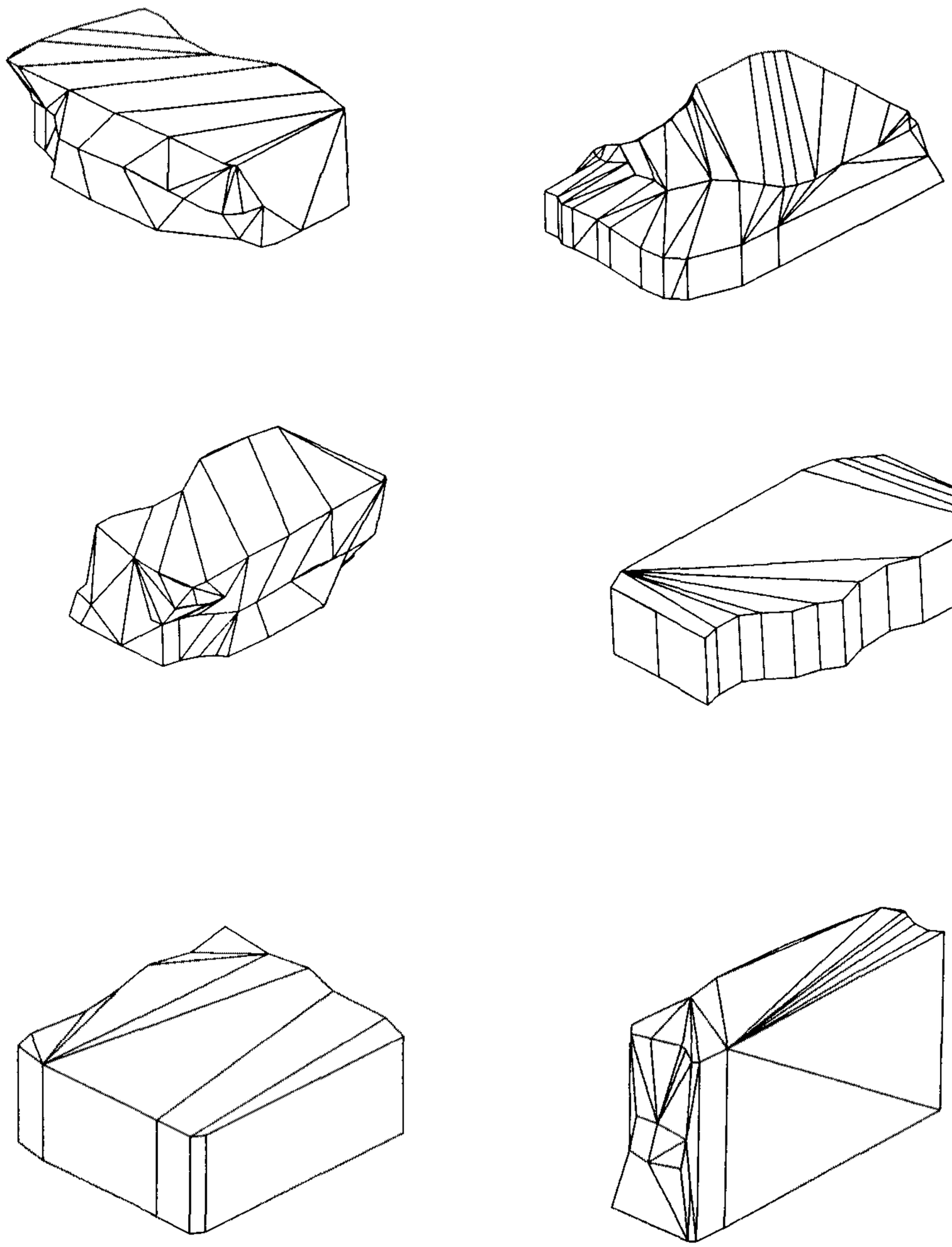
본 연구는 [그림 4-6]과 같은 연구내용과 기대효과를 갖고 있으며, 효율적인 연구를 위하여 1차,2차로 나누워서 이루워졌다.

1차작업은 미국에서 개발한 Visualization S/W인 Wavefront와 자체 개발한 S/W의 지형자료 변환모듈을 함께 이용하여 [표 4-1]의 시나리오에 따라서 약 3분짜리 애니메이션을 제작하는 작업이었다.

[표 4-1] 미륵사 서탑 복원 애니메이션 시나리오

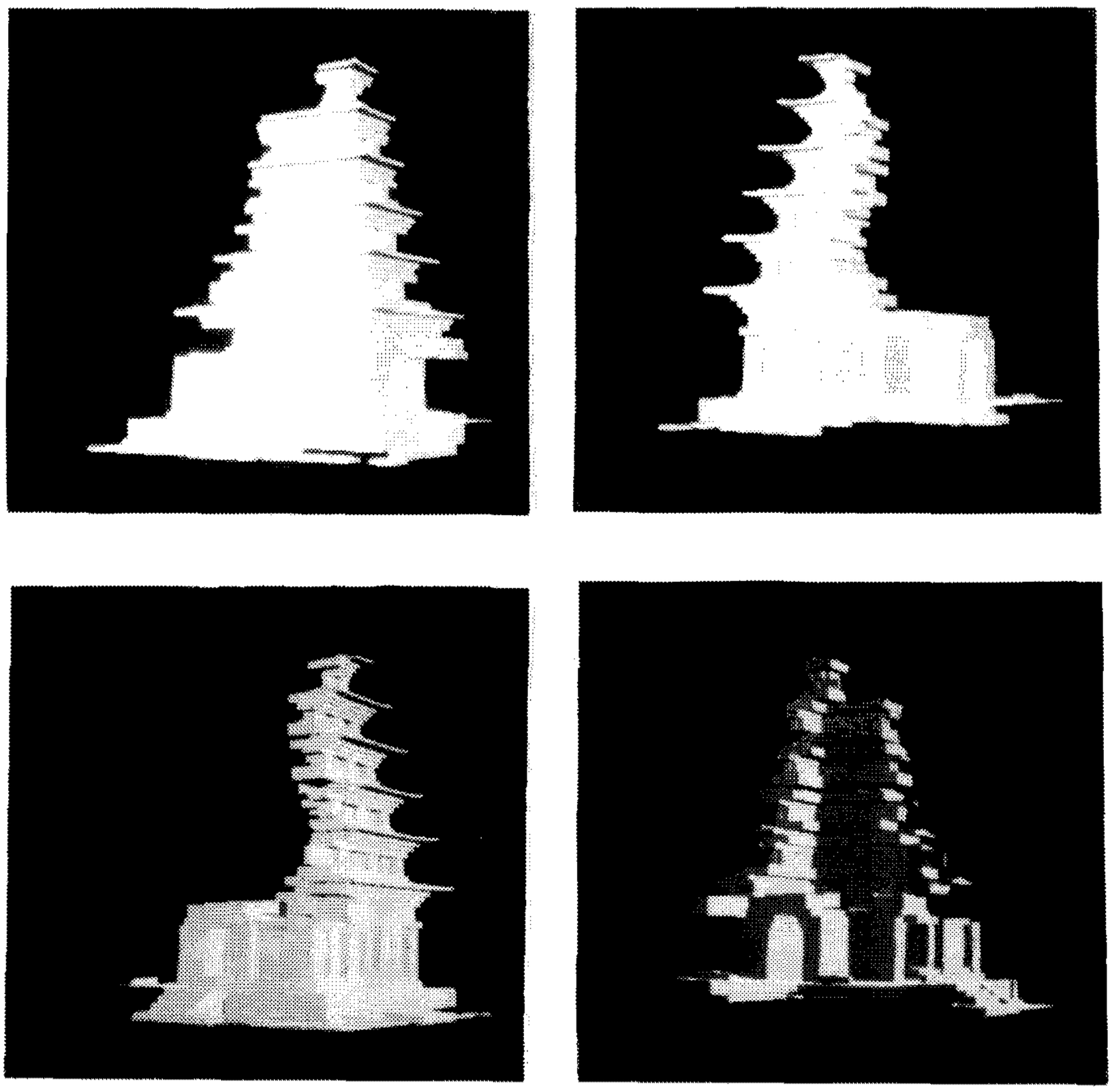
SEQ-1	시점이동(20초) o 지형 + 미륵사지 배치도 + 구름 o 잔존부재 Fade In o 잔존부재 조망 o 없어진 부재 Fade In & 동측에서 북측으로 이동 o 전체 조망	 5초 2초 5초 4초 4초
SEQ-2	시점회전(64초) o 지형 + 미륵사지 배치도 o 1층에서 4층까지 쌓아올리면서 Zoom Up o 낙석부재가 바닥에서 날아서 맞추어 봄 o 5층에서 9층까지 쌓아올리면서 Zoom Down	 2초 26초 14초 22초
SEQ-3	시점전후이동(18초) o 전체 조망 o 우측에서 좌측으로 회전하면서 이동 o 부재의 재질을 진안석으로 변경 o 부재의 재질을 가평석으로 변경 o 전체 조망(원래색으로 환원)	 3초 3초 3초 3초 3초
SEQ-4	시점회전(48초) o 지형 + 미륵사지 배치도 o 7층형으로 부재별로 쌓아 올라감	 2초 46초
SEQ-5	시점회전(20초) o 전체 조망 o 탑위로 시점 이동 o 탑 주위를 돌면서 하강 o 전체 조망	 5초 2초 8초 5초

1차작업을 수행함으로써 Wavefront는 렌더링 작업시간이 많이 소요된다는 단점과 유명 S/W 답게 User Interface가 잘되어 있다는 장점을 발견할 수 있었다.

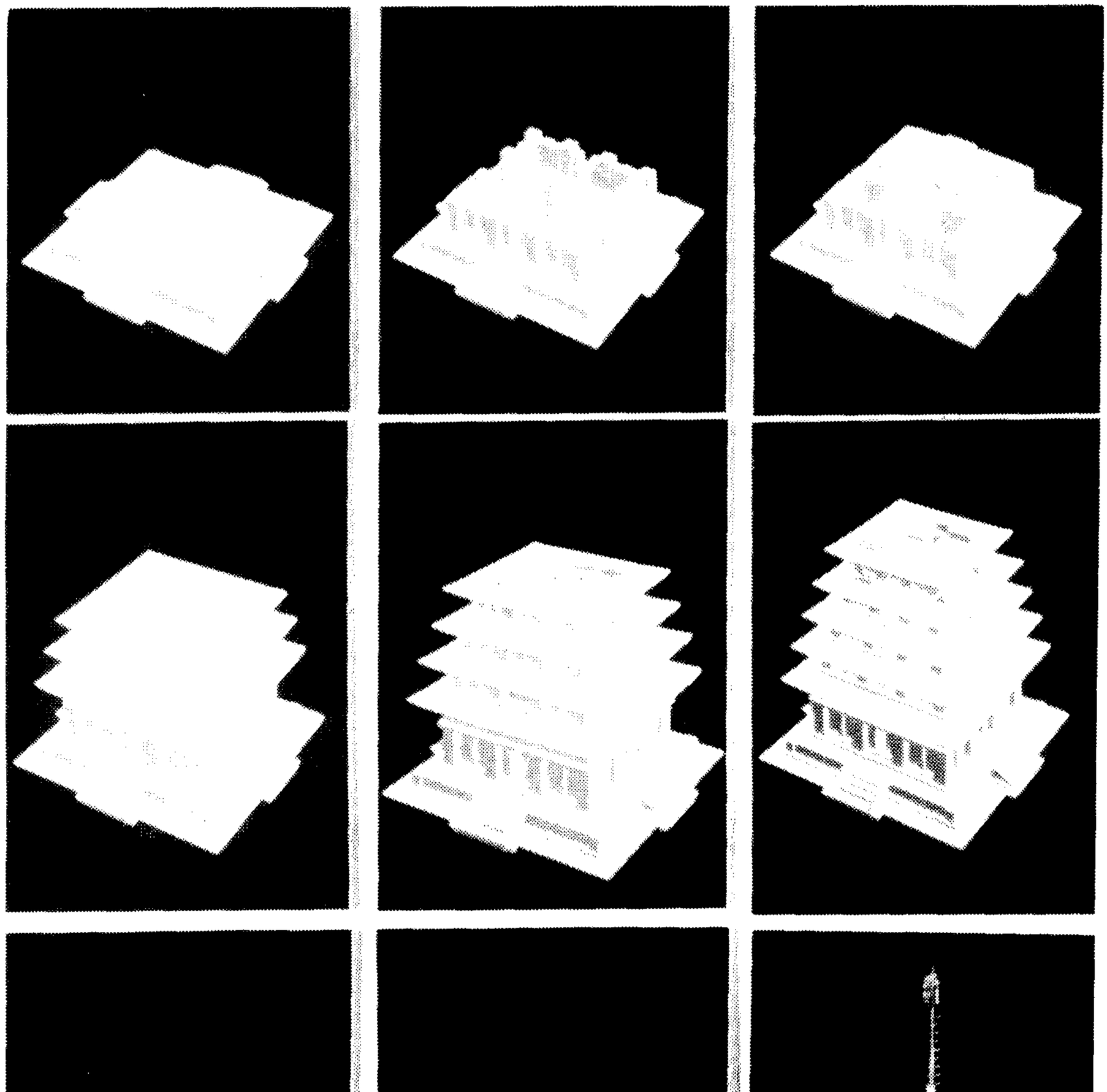


[그림 4-7] 낙석부재 위치추정을 위한 형상데이타

2차작업은 자체 개발한 S/W 만을 이용하여 낙석부재의 위치추정을 하였다. 낙석부재의 위치추정 작업을 위해서는 [그림 4-7]과 같이 부재들의 형상데이타를 원래모습에 맞추어 모델링하고, 낙석부재가 있었던 빈 공간과의 관계를 시뮬레이션하여 추정결과를 문화재 전문가의 고증자료로 제시하였다. [그림 4-8]은 잔존하는 미륵사지 서탑을 동,서,남,북 4방향의 View를 표현한 화면이고, [그림 4-9]는 미륵사지 서탑이 조립되는 과정을 나타낸것이다.



[그림 4-8] 미륵사지 서탑의 4방향 View



[그림 4-9] 미륵사지 서탑의 조립되는 과정

제 5 장 결론

본 연구과제의 결과물이 상업적으로 성공한 외국의 Visualization 소프트웨어보다 모든 면에서 우위를 갖도록하기에는 현실적으로 한계가 있다. 이와같은 이유에서 본 Visualization 시스템은 사용자가 편리하게 사용할 수 있는 환경을 제공하는것과 렌더링 속도 향상등의 두가지 측면에서 외국의 유명 소프트웨어와 비교 우위를 갖는 소프트웨어를 개발한다는 목표를 갖고 본 연구과제를 수행하였다.

렌더링 성능 향상을 위하여 외국 유명 소프트웨어를 분석한 결과 기존의 외국의 소프트웨어의 경우는 하드웨어 간의 호환성을 고려하여 그래픽 워크스테이션이 갖고 있는 그래픽 엔진의 성능을 활용하지 않고 대부분의 중요한 렌더링을 CPU의 성능만으로 해결하고 있다.

따라서 본 연구에서는 처음부터 기존의 외국 소프트웨어가 가지고 있는 범용성에서 탈피하여 우선적으로 건축분야에의 활용에 촛점을 맞추어 시스템 설계를 하였고, 그래픽 워크스테이션의 그래픽 엔진의 성능을 최대한 발휘할 수 있는 프로그램을 작성하였다. 그 결과 기존 소프트웨어 렌더링 성능의 10배 이상의 속도로 렌더링할 수 있는 시스템을 구축할 수 있었다.

최근의 컴퓨터 그래픽 분야의 기술발전 주기는 6개월 이하로 빨라지고 있다. 컴퓨터 하드웨어의 발달 또한 앞으로 2-3년내에 1000MIPS의 CPU가 개발되어질 것으로 예상되어지고 있다. 이러한 환경의 변화에 대응하기 위해서는 컴퓨터 그래픽 분야의 표준화 경향과 하드웨어 간의 호환성을 고려한 시스템 개발을 하지 않으면 안된다.

본 연구에서 사용한 그래픽 라이브러리인 starbase는 비록 그래픽 표준은 아니지만 현재의 그래픽 표준인 PHIGS 개발의 기본이 된 CGM (Computer Graphics Metafile)에 준거하여 작성된 라이브러리이며 HP사에서도 Starbase 와 PHIGS를 통합시킨 새로운 라이브러리를 개발하여 곧 공개할 예정이다. 따라서 새로운 그래픽 라이브러리가 등장하더라도 기본 구조의 변경없이 시스템을 재구성할 수 있을 것이다. 또한 본 연구에서 활용한 Graphics User Interface Tool인 OSF/Motif는 워크스테이션 분야에서의 X-window프로그램 개발을 위한 표준 tool로 정착 되어 가고 있고 PHIGS와 X-window를 병합시킨 PEX (PHIGS Extention to X-window)가 새로운 표준화 안으로 결정되어 각사에서 implementation 작업을 진행중이다. 따라서 본 연구에서 사용된 그래픽 라이브러리나 Graphics User Interface는 차세대 표준화 안을 최대한 수용할 수 있는 것으로서 앞으로의 시스템 환경 변화에 충분히 적응할 수 있을 것으로 사료된다.

참 고 문 헌

1) J. Foley : Computer Graphics, Principles and Practice, 2nd ed., Addison Wesley, 1990.

2) A. Watt : Fundamentals of Three-Dimensional Computer Graphics, Addison Wesley, 1989.

3) D. Thalmann : Scientific Visualization and Graphics Simulation, Wiley, 1990.

4) Ed. by R. H. Mendez, Visualization in Supercomputing, Springer, 1990.

5) G. Bancroft, T. Plessel, F. Merrit, V. Watson: "Tools for 3D Scientific Visualization in Computational Aerodynamics", NASA Conf. Publ., NASA-CP-3045, p.55,1989.

6) 横田秀明, 橋本明彦, 西村徹: "サイエンティフィック・ビジュアリゼーションの現狀", 富士總研技報, 1, no.2, p.233, 株式會社富士總合研究所, 1990年12月1日.

7) 藤井孝藏, "流體解析のためのビジュアライゼーション", PIXEL, no.100, p.88, 1991.

8) B. H. McCormick, T. A. DeFanti, M. D. Brown(ed.): "Visualization in Scientific Computing", Computer Graphics, 21, no.6, Nov.1987.

9) K. A. Winkler, J. W. Chalmer, S. W. Hodson, P. R. Woodward, N. J. Zabusky: "A NUMERICAL LABORATORY", Physics Today, 40, no.10, p.28, 1987.

10) L. J. Rosenblum: "Scientific Visualization at Research Laboratories", IEEE Computer, 22, no.8, p.68, 1989

11) "グラフィックス・スーパーとネットワーク技術がビジュアライゼーションを變革", 日經CG, no.32, p.87, 1989年5月.

12) "低價格化・高性能化するワークステーション", PIXEL, no.95, p.100, 1990.

13) “IBM本格参入により活氣づくスーパーグラフィックス・ワークステーション”, 日經CG, no.46, p.15, 1990年5月.
14) “パーソナル3次元グラフィックスの夢をかなえるi860の世界”, 日經CG, no.46, p.15, 1990年7月.
15) R. Stock: “The Super Chip: Looking under the hood of Intel's i860 graphics chip”, COMPUTER GRAPHICS WORLD, no.7, p.85, 1990.
16) “i860とカスタムLSIの採用で100万ベクター/秒を達成した HP Turbo VRX”, 日經CG, no.49, p.164, 1990年10月.
17) “ビジュアライゼーションに性能を發揮するPOWERVISIONの新技術”, 日經CG, no.47, p.146, 1990年8月.
18) D. R. Baum, J. M. Winget: “Parallel Graphics Applications”, UNIX REVIEW, 8, no.8, p.50, 1990.
19) クボタコンピュータメーカ資料(901015N-T 009.1), Graphics Supercomputer TITAN 3000 Vシリーズ, クボタコンピュータ株式會社.
20) 姫野龍太郎: “コンピュータ・シミュレーションによるセフィーロ周りの空氣流”, 日經CG, no.46, 1990年7月.
21) 日立メーカ資料(SW-788), 日立スーパーコンピュータHITAC S-820動畵像出力システム,株式會社日立製作所, 1990.3.
22) 大島務, 横田秀明, 橋本明彦, 西村徹, 小池秀躍, 中川哲夫, 千葉幾子 : “スーパーコンピューティング・システム・インテグレーションの現狀と展望”, 富士總研技報, 1, no.1, p.146, 株式會社富士總合研究所, 1990年6月1日.
23) Ed. by R. H. Mendez, Larry Smarr: “Supercomputing Environment for the 1990s”, Visualization in Supercomputing, Springer, pp.184-202, 1990.
24) Cray maker 資料 (MCSW-2-389), MPGS Multipurpose Graphics System, Cray Research, Inc.
25) “機能分散處理による多目的グラフィックシステムMPGS”, in Cray maker 資料 CASME 90 ,日本Cray株式會社, pp.77-94, 1990

26) 白山晋，桑原邦朗，“計算流體力學における可視化技術とその問題點”，日本物理學會誌，45，no.7，p.483，1990.
27) 藤井孝藏，PIXEL，no.95，p.92，1990
28) 角 文雄，“コンピュータ・グラフィックス動畵システムCGMSの紹介”，PIXEL，no.87，p.107,1989.
29) Lerner，D.，林 伸彦，“世界初全天周フュルカラ-立體映像ユニバース2-太陽の宴”，PIXEL，no.92，p.163，1990.
30) C. Upson, T. Faulhaber, Jr. D. Kamins, D. Laidlaw, D. Schlegel, J. Vroom, R. Gurwitz, A. van Dam: “The Application Visualization System : A Computational Environment for Scientific Visualization”, IEEE Computer Graphics & Applications, 9, no.4, p.30, July 1989.
31) “SIGGRAPH ビジュアリゼ-ション最前線”，日經CG，no.49，p.121，1990年10月.
32) G. Freiherr, G. Pfitzer: “The New Breed, Today's visualization toolkits emphasize ease of use for nonprogrammers”, COMPUTER GRAPHICS WORLD, 14, no.1, p.44, 1991.
33) D. L. Brittain: “Portability of Interactive Graphics Software”, IEEE Computer Graphics & Applications, 10, no.4, p.70, July 1990.
34) 河內隆幸，“SIGGRAPH'90報告(1)”，PIXEL，no.97，p.49，1990.
35) C. Upson: “Tools for Creating Visions”, UNIX REVIEW, 8, no.8, p.39, 1990.
36) D. S. Dyer: “A Dataflow ToolKit for Visualization”, IEEE Computer Graphics & Applications, 10, no.4, p.60, July 1990.
37) M. Hess: “Flex PEX: Exercising 3-D Graphics in a Distributed Environment”, SUN TECH JOURNAL, 3, no.1, p.71, Winter 1990.
38) R. J. Rost, J. D. Friedberg, P. L. Nishimoto: “Xウィンドウを擴張し,3次元グラフィックス機能を擴張したPEX”,日經エレクトロニックス，no.491，p.199，1990.1.22.

39) “Xウィンドウの3次元擴張版,PEXの全容が明らかに”, 日經CG, no.47, p.99, 1990年8月.
40) “Xウィンドウがリリース5で飛躍する”, 日經エレクトロニックス, no.510, p.103, 1990.10.1.
41) The X Window System series, (Vol.0, 1, 2, 3, 4, 5, 7), O'Reilly & Associates.
42) “Xウィンドウ端末,文字端末からワークステーション,グラフィックス裝置へと廣がる”, 日經エレクトロニックス, no.499, p.155, 1990.5.14.
43) H. C. K. Sung, G. Rogers, W. Kubitz: “A Critical Evaluation of PEX”, IEEE Computer Graphics & Applications, 10, no.6, p.65, November 1990.
44) 佐藤弘廣, 石畑明, 波形肇, “高竝列計算機CAP-IIによる三次元グラフィックス”, 電子情報通信學會研究報告, 90, no.144, pp.133-138, 1990.
45) “最大性能12.8GFLOPSの竝列コンピュータを開發中”, 日經エレクトロニックス, no.499, p.155, 1990.5.14.
46) Ed. by R. H. Mendez, H. Sato, M. Ishii, M. Ikesaka, K. Murakami, H. Ishihata: “Cellular Array Processor CAP and Visualization”, Visualization in Supercomputing, Springer, p.100, 1990.
47) Computer Graphics World, 13, no.9, 1990.
48) A. T. Pang: “Line-Drawing Algorithms for Parallel Machines”, IEEE Computer Graphics & Applications, 10, no.5, p.54, September 1990
49) Introduction to CM Visualization, Version 5.2, Thinking Machines Corporation, September 1990.

부　　록

Graphical User Interface

여 백

목 차

그 림 목 차

표 목 차

Graphical User Interface

기존의 UNIX 환경하에서는 User Interface를 지원하기위해 Command line 방식을 사용했다. 사용자는 Unix Shell 언어와 같은 것을 익혀야했고, 시스템과 키보드를 통하여 의사소통을 해야하므로 여러가지 문제점이 대두되었다. 특히 키보드 입력으로는 초당 10 개 정도의 문자 밖에 입력할 수 없고, Unix Command 하나를 위해 너무 긴 명령어나 옵션을 사용할 수 없으므로 결국은 명령어나 옵션 자체가 암호화 되어 버리는 결과를 가져 왔다. 또한 Unix의 명령어 구조가 매우 명백하고, 일관성있는 구조를 가지고 있다고는 하지만 사용자에게 암호화된 많은 것 들을 기억하도록 강요하는 평면구조의 명령어 이기 때문에 초보자 들은 많은 어려움을 겪어야 했다. 비록 이러한 문제점을 돕기위해 man 이 제공되고있지만, 설명이 매우 길며, 문법적 서술로 되어있고, 또한 사용자가 그 설명에 관한 결과를 거의 알고 있어야 하기때문에 효과적인 도움을 주지 못하고 있다. 그러나, X Window System을 계기로 Unix 상에서의 User Interface를 GUI(Graphical User Interface)로 전환하게되면서 이러한 비능률적인 사용자 인터페이스가 제거되기 시작하였다.

Unix의 초보 사용자들의 User Interface를 쉽게 해주고 프로그래머에게 일관성있고, 쉽게 사용자 인터페이스를 만들 수 있도록 MIT와 HP 등이

X Tooklit을 개발하기위한 Project를 시작하여 결국은 Widget Set이라는 강력한, Graphical User Interface Component를 개발 하였다. Widget Set으로 특히 응용 프로그램의 User Interface에 있어서 가장 Primitive 한 Component(예를 들어 PushButton, Menu, ScrollBar 등)를 쉽게 작성 할 수 있게 되었고, 프로그래머의 기호에따라 새로운 Widget을 개발하기 쉽도록 되어 있으며, Widget에 대한 Action을 필요에 따라 프로그램 내에서 쉽게 바꿀 수 있도록 되어 있기 때문에, Widget Set을 사용하여 User Interface를 개발하는 프로그래머는 특정한 스타일에 제약을 받지 않고 융통성이 있으면서도 일관성 있는 프로그램을 작성할 수 있게 되었다. 이와같이 서로 충돌을 일으키기 쉬운 융통성과, 일관성, 용이성이 잘 조화되어 프로그램밍에 있어서 새로운 차원을 제공하고 있는 것이 바로 X Toolkit이다.

복잡한 현대 사회에서 소프트웨어의 복잡도는 증가 되고 있고, 소프트웨어의 수명은 점점 짧아지고 있다. 이에따라 요구되어지는 소프트웨어 개발기간은 더욱 단축되고 있기 때문에 이와같은 문제를 해결하기위해 소프트웨어 공학자들은 소프트웨어의 개발, 작동, 유지, 폐기에 관한 새로운 접근법 들을 시도하게 되었다. 즉 노동집약적인 생산 방식과 이에따른 낮은 소프트웨어 생산성, 개발되는 소프트웨어의 품질문제, 인력 부족등을 극복하기위한 노력을 하게되었고, 소프트웨어의 재사용, 이식성, 관리

성(maintainability)을 중요한 요소로 부각 시키게 되었다.

X Toolkit은 그 자체가 소프트웨어 재사용의 개념을 배경으로하고 있으며, X Window System과 X Protocol은 Portability의 문제를 해결해 주고 있다. 또한 X Toolkit을 사용하므로써 프로그램의 융통성, 일관성, 용이성이 좋아지기 때문에 프로그래머는 프로그램 개발에 드는 비용과 기간을 단축시킬 수 있고, 빠른 Prototyping이 가능하게 됨으로써 소프트웨어의 수요자 요구사항에 맞추어 소프트웨어를 쉽게 개발 할 수 있게되었다. 특히 X Toolkit에서 제공되는 Widget들은 이미 testing을 끝낸 것이기에 그 만큼 더 프로그램의 정확도 검사를 위해 투여해야 하는 시간, 노력을 줄일 수 있다. 결국 많은 융통성을 지닌 high-level의 Toolkit 사용은 프로그래머의 생산성을 크게 향상시키는 방편이 될 수 있으며, 고급의 User Interface를 소프트웨어의 사용자에게 제공할 수 있는 길이기도 하다.

이 장에서는 먼저 Graphical User Interface 개요 및 정의, 사용자 인터페이스 요소, GUI 디자인 원칙 등을 설명했고, X Window 의 개념과 Xlib, X Protocol, X의 특성, X Programming 등 X Window 시스템에 대한 전반적인 내용을 정리하였다. 특히 본 프로젝트에서 구현하는 Visualization 시스템에서는 User Interface를 Motif로 작성하였기 때문에 일반적인 Toolkit과 OSF/Motif에 대한 내용을 상세히 수록하였다.

제 1 절 Graphic User Interface 디자인

그래픽 사용자 인터페이스(Graphic User Interface)란 Window, 커서, 아이콘, 메뉴 등을 이용하여 컴퓨터와 상호작용을 하면서 사용자가 직접 조작할 수 있는 환경을 말한다. 그래픽스 패키지를 설계할 때 사용자 인터페이스를 위하여 GUI 디자이너가 고려할 점은 사용자가 시스템으로 부터 필요로 하는 정보가 무엇이며, 시스템이 사용자로 부터 필요로 하는 정보는 무엇인가, 그래픽스 시스템이 무엇을 할 수 있으며 어떤 기능이 준비되어 있는지를 쉽게 보여줄 수 있어야 하고 그래픽스 패키지가 입력과 에러에는 어떻게 대응할 것인가, 출력 표시는 어떻게 문서화하고 사용자에게 설명할 것인가도 생각해야 한다. 이 중 가장 중요한 것은 사용자 입력에 어떻게 반응 (Feedback) 할 것인가 이다. 사용자에게 정보를 어떻게 전달 하는가는 출력 형식에 의해 결정된다. 출력 화면은 사용자에게 정보를 가능한 한 가장 효율적으로 제공하도록 조작되어야 한다. 출력형식을 설계함에 있어서 고려할 요소는 물체를 표시하는데 필요한 기하학적인 패턴 들의 선택과 출력 장치 상에의 전반적인 배치 등이다.

1. 그래픽 사용자 인터페이스(GUI)의 정의

그래픽 기능을 정보교환의 매개체로하여 User Interface와 접목 시킨 것을 GUI(Graphical User Interface)라하며 사용자의 노력과 시간을 줄여 생산성을 높이기 위한 유저 인터페이스로서 다음과 같은 중요한 역할

을 한다. 첫째, 복잡한 문제를 직관적으로 해독, 판단 할 수 있게 해 준다. 둘째, 소프트웨어의 도처에 나타나는 User Interface 구성요소가 같은 형태로 이루어짐으로써 일관성을 부여해 준다. 셋째, 사용자가 직접 느껴서 원하는 Interface를 선택할 수 있는 시각적 단서를 제공한다. 넷째, User Interface의 색상, 아이콘, 키보드, 마우스 사용에 이르기까지 사용자의 취향 습관에 따라 바꿀 수 있는 유연성을 제공한다.

소프트웨어의 설계에 있어서, 사용자와 컴퓨터간의 정보 교환을 위한 매개체라 볼 수있는 User Interface를 취급하는 코드 부분이 전체의 50-80% 에 달하고 있으며, 또한 User Interface의 발전은 소프트웨어의 질을 좌우 할 만큼 비중이 높아지고 있다. 사용자의 입장에서 볼때, high-level User Interface는 사용자가 소프트웨어를 사용할 때 소비하는 노력과 시간을 줄여 생산성을 높이는데 중요한 역할을 한다.

2. 사용자 인터페이스의 요소

가. 사용자 모델 (User Model)

사용자 모델은 시스템이 무엇을 하도록 설계되어 있는가와 어떤 그래픽스 작업을 할 수 있는지를 보여 준다. 예를 들어 건축설계를 위한 그래픽스 패키지를 만든다면 패키지의 작업들은 문, 창등의 건물 요소를 지정된 위치로 이동시킬 수 있어야한다. 전반적으로 사용자 모델은 단순하고 일관성이 있어야 한다.

나. 명령 언어 (Command Language)

유저 인터페이스는 가능한한 사용자의 암기를 최소화하고, 초보자라도 사용할 수 있도록 사용자 도움말 기능을 두는 것이 매우 중요하다. 어떠한 일련의 조작 기능에는 작업의 실행이 완료되기 전에 취소되기도 하고, 작업이 시작되기 전의 상태로 복귀될 수 있는 기능이 있어야한다.

다. Feedback 방법 (Feedback Method)

그래픽스 패키지는 사용자 입력에 대한 즉각적인 반응을 약 0.1초 안에 하도록 설계되어야 한다. 메시지는 간결해야하며 진행중인 처리의 유형을 명확히 표시해야 한다. 즉 작업중일 때는 번쩍거리는 메시지를 보낸다든지 메뉴를 반전하여 처리되고 있는 메뉴를 보여주는 방법이 있다. 처리 시간이 0.1초 보다 길어질 경우 즉각적인 응답으로 사용자에게 단순히 입력이 받아들여졌다는 것을 알려 주거나 단순히 에코를 출력하는 방법도 있다.

라. 출력 형식 (Output Format)

사용자에의 출력형식을 배치하고 제공하는 방법은 수없이 많기 때문에 최대의 시각효과를 얻기위해 가장 알맞은 출력형식을 어떻게 설계할 것인지 고려해야 한다. 출력 형식 설계에서는 메뉴와 메시지 구조, 아이콘 및 심볼의 모양, 전반적인 화면의 구성이 포함된다.

아이콘 및 심볼의 모양은 패키지의 응용 유형에 따른다. 심볼 모

양은 단순하면서도 표시하려는 물체나 작업을 명확하게 나타내야 한다. 화면 구성의 세가지 기본 요소는 사용자 작업 영역, 메뉴 영역, Feedback 영역이다.

3. GUI 디자인 원칙

가. 일관성 (Consistency)

입출력 부분에서 일관성이 있을 수 있는데, 출력 부분에서 같은 코드를 언제나 쓴다 든지, 시스템 메시지가 언제나 정해진 부분에 나온다 든지 하는 것이고, 입력 부분에서는 리턴 키, 탭, Backspace 등의 키가 언제나 정해진 기능을 수행한다 든지, Help 같은 명령어를 아무 때나 부를 수 있고 Move, Copy, Delete등 누구나 예상하고 있는 대로 작동하여야 하고, 메시지는 일정한 영역에 나타나는 것 등을 말한다.

1) 외부적 일관성

사용자가 여러가지의 윈도우 시스템 사이에 혹은 동일한 윈도우 시스템에서 응용프로그램 사이의 일관성을 느낄 수 있도록 디자인 하는 것을 말한다. 외부적인 일관성의 유지로 인하여 사용자는 다양한 윈도우 시스템 사이를 오가며 사용할 경우에도 친숙함을 느낄 수 있다.

2) 내부적 일관성

표준화된 요소를 사용함으로서 사용자에게 확신을 줄 수 있도록 GUI의 모든 구성요소 들에 대한 관례와 규칙을 지킨다.

나. 안정성 (Stability)

사용자들은 컴퓨터 환경이 유동적으로 바뀔 때보다 그것을 이해할 만하고 친근할 때 편안함을 느낀다. 안정성을 위해서는 GUI 디자인을 할 때 일관성 있는 개념적 구조를 제공해야 하며 사용자는 지침서를 잘 따라야 한다. 안정성이 있는 GUI를 통해서 사용자는 한번 대상을 다루어 본 후 다른 대상에도 같은 방법을 적용 시킴으로써 작업 효과를 높일 수 있다.

다. 심미성 (Aesthetic Integrity)

효율적인 사용자 인터페이스를 위해서는 표시물이 시각적으로 아름다우며 혼동스럽지 않아야한다. 또한 다른 대상물은 화면에서도 다르게 모여야 하며, 심미성의 문제는 개인차가 있으므로 표현하려는 대상물이 보기에 좋으면서도 그 내용이 적절히 표현되어야 한다.

라. 간결성 (Simplicity)

디자이너가 커뮤니케이션을 위하여 꼭 필요한 요소만을 포함시켜 디자인 하는 것을 의미한다.

마. 명료성 (Clarity)

혼란을 일으킬 만한 요소를 최소화시켜 표현하려는 정보의 의미를 명확히 하는 것을 말한다.

제 2 절 X-Window 시스템

Unix 환경의 다양한 워크스테이션을 하드웨어에 상관없이 네트워크로 연결하여 사용할 수 있도록 하기위해서 1984년 MIT에서 프로젝트 Athena를 시작하였다. 이 Project는 서로 다른 하드웨어에 대해서도 높은 호환성을 갖는 시스템, 대학내의 어느 시스템에서도 필요로하는 다른 시스템 자원을 접근할 수 있는 컴퓨팅 환경을 학생들에게 제공하기 위한 네트워크에 기반을 둔 차세대 교육용 Workstation-Computing 시스템 개발로 사작되었다. 이 프로젝트의 결과로 X-Window 시스템이 개발되었는데 이는 네트워크을 통하여 다른 컴퓨터 시스템을 불러 마치 자신의 시스템을 이용하는 것 처럼 윈도우에 Bit-mapped 그래픽으로 디스플레이 할 수 있는 그래픽스 윈도우 시스템이다. 각종 워크스테이션에서 이 X-Window 시스템을 쓸 수 있기 때문에 X-Window의 이용은 급속히 확산되었고, 이에 MIT에서는 1987년 기업, 연구소 등과 함께 X-Window 시스템을 관리 향상시키기 위하여 X 콘소시엄 (Consortium)을 결성하였다. 1987년에 X11, 1988년후반에 X11R3을 발표한 MIT는 1990년 1월 X11R4를 발표하였고, 3D 디스플레이 기능을 갖춘 PEX (PHIGS Extensions to X)를 지원하는 X11R5도 이미 미국의 대학교에서 사용되고 있다. X-Window 시스템의 최대 장점은 디바이스 독립성(Device Independency)과 네트워크 투과성(Transparency)으로서 네트워크에 연결된 어느 Workstation에서도 동일한 User Interface를 가능하게 해준다. 그리하여 현재 많은 Vendor들이 이 X-Window 시스템을 표준 Interface로 채택하는 경향이 있다.

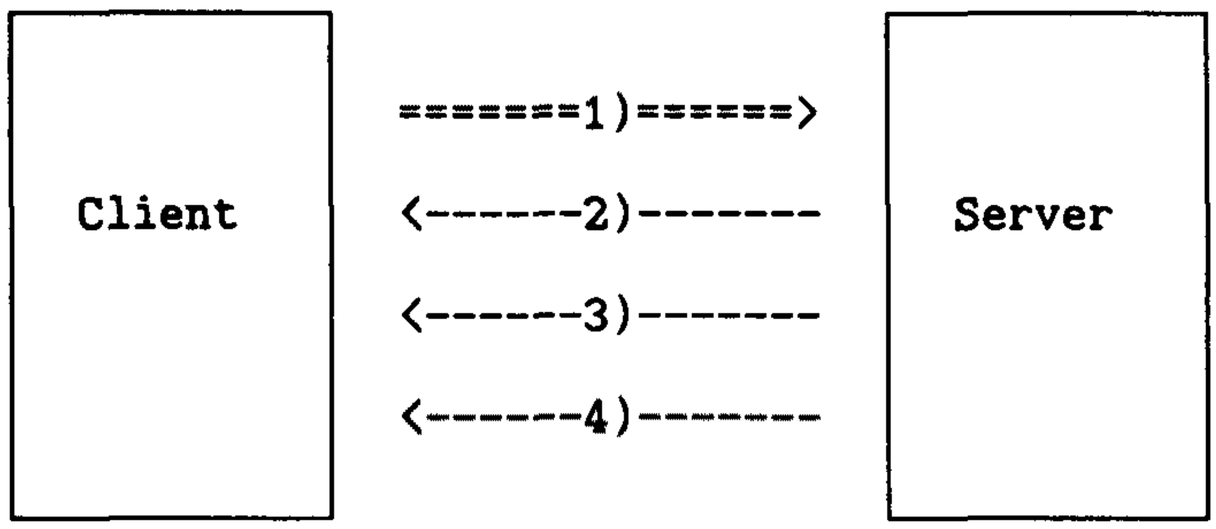
1. X Window System의 기본 개념

가. X Protocol

X Window System은 응용 프로그래머에게, 이식성이 높은 GUI(Graphical User Interface)를 제공하는 산업표준의 Software System으로, 상이한 기종의 machine에서도 X Window System이 장착된 경우에는 사용자에게 동일한 환경을 제공한다. 과거 80년대의 주류를 이루던 단말장치가 주컴퓨터와는 단지 저해상도의 화면상에서 문자와 숫자로만 통신이 가능했다면, Computer Graphic, 고 해상도 Bitmapped Video Display, Keyboard, Mouse를 하나로 집결시킨 Workstation은 90년대의 컴퓨터 사용자에게 더욱 친밀한 컴퓨팅 환경을 제공해 주고 있다고 볼수 있다. 그러나 Hardware에 따라 서로 상이한 Graphical Interface를 개발하는데 중복투자 되는 문제점 들을 갖고있다.

이러한 문제점을 X 는 응용 프로그램들과 하드웨어 사이에 X Protocol 이라는 중간 완충 지대를 이용하여 해결해 주고 있다. 이러한 면에서 볼때 X Protocol은 Open System에서 말하는 이른바 Software Platform으로써의 기능을 한다고 볼 수 있다.

X Protocol은 양 방향 통신이 가능한 lower-level 네트워크 프로토콜 위에서 동작한다. TCP/IP 와 DECnet이 현재 X Server에 의해 지원되는 가장 일반적인 lower-level의 Network Protocol이다.



[그림 A-1] X Protocol

즉, X Protocol 이란 Client 와 Server 간의 통신을 위한 기본 메시지로서 그림 [A-1]과 같이 4가지 종류가 있다.

1) Request

Client가 Server에게 Window 생성이나, 이동, 그래픽 이미지 출력등의 요구를 할때 사용된다. Request Message는 Server에 바로 전달되지 않고 Xlib 버퍼에 저장 되었다가 Buffer가 꽉 차거나, Server로부터의 Reply가 요구되는 경우에 버퍼가 Flush되고, Server에 Request 들이 보내지게 된다. 즉 이러한 버퍼 Mechanism 을통해 Network load를 낮춤으로써 시스템 전체의 성능이 극대화 된다.

2) Reply

대부분의 Request는 One-Way Protocol Message로 "Send-and Forget" 법주에 속한다. 그러나 때로는 XQueryPointer(); 와 같은

Xlib 루틴은 Round-Trip Request Message를 생성하여, Client로 하여금 Server로 부터의 응답을 받을때까지 Blocking상태에 놓이게한다. Round-Trip Request Message는 버퍼를 Flush하여, 저장된 Request를 Server에 전송한 후 자신도 바로 Server에 전송되고 Server는 요구사항을 처리한 후 그 결과를 바로 해당 Client에게 보낸다.

3) Event

Server가 관리하는 machine에서 발생한 Event를 알려 준다 (KeyPress, MouseButtonPress,Window Size 변경등).

4) Error

Server가 Error발견시 Client에게 알려준다.

나. Client-Server 모델

대부분의 다른 Window System들은 Kernel-Based이다. 즉 특정 OS에 매우 밀착 되어있어서 단지 한개의 한정된 시스템 안에서만 이용이 가능한 반면 X Window System은 OS의 일부분이 아닌, 완전히 User-Level의 프로그램으로 구성되어 있고 Client-Server Model에 기초하고 있다. 응용 프로그램이 한 Machine에서 실행되고, 그 결과는 다른 Machine에 출력할 수 있도록 하기위하여, X는 2개의 Process간의 Network Protocol을 이용한다.

1) Client

이들 프로세스중 X Server에 의해 제공되는 기능들을 이용하는 응용 프로그램을 Client라 한다. Client는 윈도우 생성을 요구하거나 도형의 출력 등을 Server에 요구하며 이요구는 X Protocol 이라는 Nerwork Protocol에 의해 Server에 전송된다. 이와같이 Client는 Device와 직접적으로 입출력을 행하지는 않는다.

2) Server

X Server는 키보드나 마우스 등을 사용한 사용자의 입력과 응용프로그램의 Request를 처리하여 화면에 출력하는 기능을 맡은 프로세스를 Display Server 혹은 간단히 Server 라고 한다. Server의 전체 또는 일부가 Software가 아닌, Hardware나 Firmware 로 구현된 것을 X Terminal 이라고 한다. Server는 하드웨어 간의 상이성을 감추며, 다음과 같은 일 들을 수행한다.

가) 여러 Client가 Display를 Access할 수 있도록 해준다.

(Server에 의해 관리되면서 Screen상에 나타난 사용자 프로그램이 Client이다. 하나의 Server에 여러개의 Client들이 connect 될 수 있다.)

나) Client로부터의 Network Message(Request)들을 해석해서, 그에 따라동작한다. Server로 하여금 Window를 이동시키거나 그래픽 이미지를 출력하도록 요청하는 종류의 Request와 Server에게 정보를 요청하는 종류의 Request가 있다. Protocol Request는

Xlib나 X Toolkit 또는 다른 Function Library를 Client가 호출함으로써 생성된다.

다) 사용자로부터 입력을 받아 Network Message(Event)형태로 Client에게 넘겨준다. Event는 키보드나 마우스의 눌림/놓임과 마우스 움직임등이 있다.(X Server에게 인식되는 Event는<X.h>에 정의 되어있다.) Event의 발생은 비동기적으로 일어나며, 또한 여러 다른 Device에서 온 Event들이 서로 섞여있을 수 있다. Server는 각 Client에게 각각의 Event를 정확히 전송해 주어야 한다.

라) Window나 Font와 같은 복잡한 자료구조를 유지함으로써,Server는 더 효율적으로 일을 수행할 수 있다. Client는 이들 추상 자료구조를 단지 ID 번호만으로 참조할 수 있다. Server가 이와같은 추상자료구조체들을 유지하고 있음으로해서, 각 Client에의해 관리되어야하는 자료의 양을 줄이고, Server와 Client의 통신시 Network를 통해 전송되어야 하는 자료의 양을 줄여줌으로써 전체 시스템의 성능을 극대화 시키는 역활을 한다.

다. X-Window 시스템 구조

X-Window 시스템은 Client-Server 모델로 되어 있다. Client란 Request를 하는 응용 프로그램을 말하며 X-library를 Call 하는 프로그램이고 Server란 워크스테이션이나 X-터미날에서 Client의 X-lib call이 있을때 이를 수행하기 위하여 디스플레이, 키보드, 마

우스 등을 제어하는 프로그램이다.

X Window Sytem의 구조는 [그림 A-2] 와 같으며, 각 요소에 관한 설명은 다음과 같다.

1) X Server

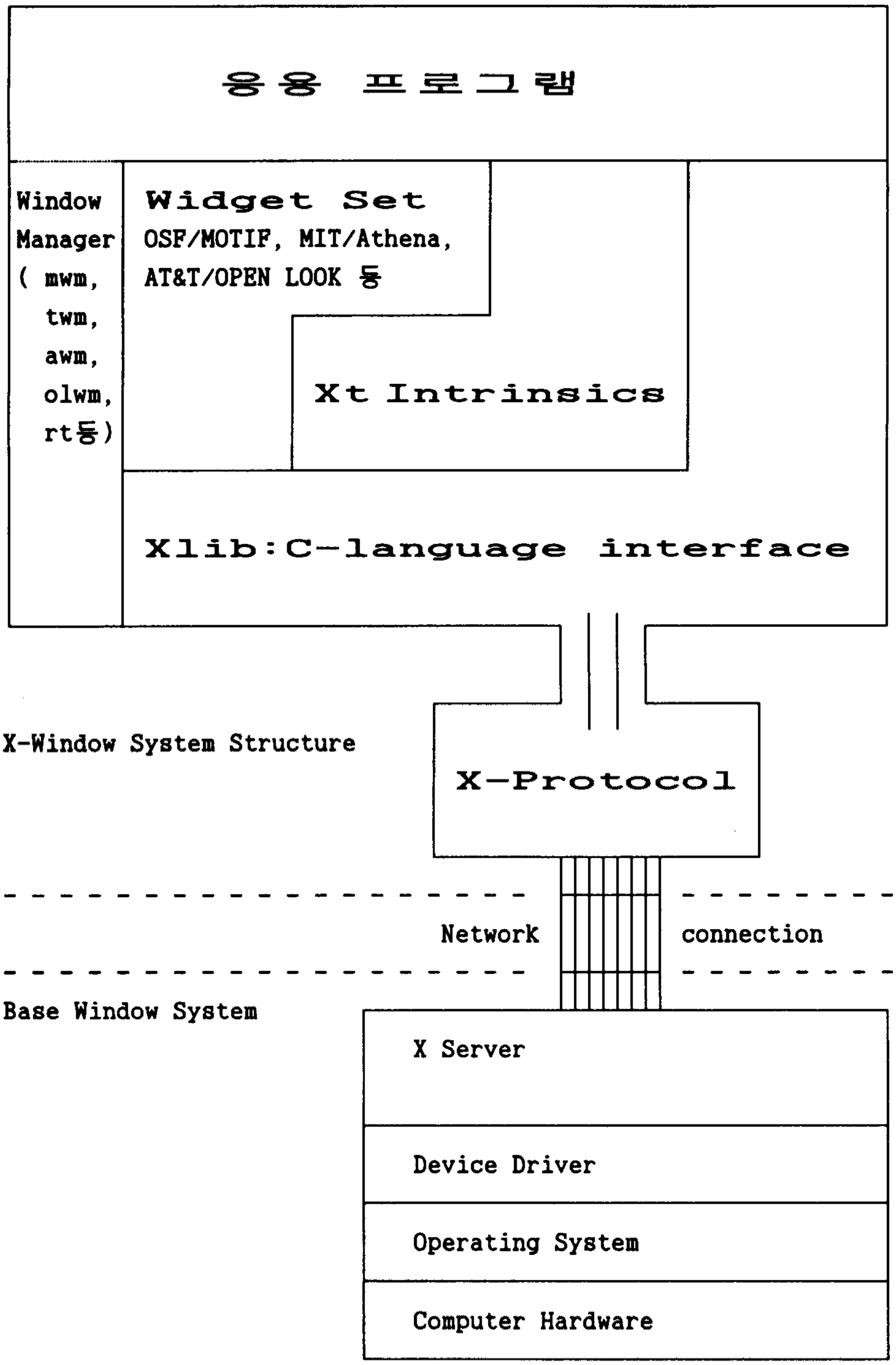
X Server는 Client의 요구에 따라 X Window 의 디스플레이, 마우스, 키보드, 스크린 윈도우 등을 제어하여 Bitmaped Graphics Operation을 수행 제공한다.

2) X Protocol

X에 Device, Vendor Independency와 Network Transparency 를 제공하기 위하여 X Protocol을 이용한다. Base Window로 Interface 할 수 있는 유일한 통로이기에, X Protocol은 Base Window의 모든 기능을 이용할 수 있도록 정의 되어 있다. X 응용프로그램은 보통 네트워크 프로토콜을 직접 이용하지 않고 프로그래밍 인터페이스를 통하여 이용한다. X 프로토콜은 Request, Reply, Event, Error 메시지로 구성되어 있으며 Client와 Server 간의 정보 교환을 최소화하고 서로 다른 기종간의 분산 시스템 환경을 지원하기 위해 OS나 하드웨어에 무관하도록 설계되어 있다.

3) Xlib

X Protocol이 Network Packet과 Byte Stream 단위에서 정의되어



X-Window System Structure

[그림 A-2] X Window System 구조

있지만, 프로그래머는 복잡한 X Protocol Command 생성을 쉽게 해 주기위해 C 언어로 작성된 Xlib 함수들을 이용해서 X_based Window 로의 low-level Interface를 한다. Xlib의 루틴 들은 다음과 같은 기능을 갖고 있다.

가) Server-Client Connection

나) Window Manipulation

다) Graphic Manipulation

라) Colormap Manipulation

마) Event Manipulation

바) Mouse Manipulation

사) Keyboard Manipulation

그러나 프로그래머가 Xlib로 직접 프로그램하는 것도 상당히 어려우므로 그위에 프로그래머의 노력을 덜어줄 수 있는 Widget Set 과 Xt Intrinsics, X Toolkits 등이 등장하였다.

4) Toolkits

가장 보편적으로 많이 쓰이고 있는 X Toolkit은 보통 Widget Set 과 Xt Intrinsics로 나뉘어져 있으며, Object Oriented Programming (OOP) Style로 작성되어 있다. 즉 기초 블럭이 Widget이며 Scrollbar, Titlebars, Menus, Dialog Boxes 같은 User Interface Components 들이 부품(Object)이고 프로그래머

가 의도한 User Interface를 만들 수 있도록 하기 위해서, User Interface Component 들을 조합할 수 있도록 해주는 Framework를 제공하는 것이 X Intrinsics이다.

라. X 의 특성

X 의 성격내지 특성을 간략히 표현하면 다음과 같다.

1) 네트워크 투과성 (Network Transparency)

한 Local Machine에서 필요에 따라 응용프로그램을 Remote Machine에서 실행시키고 그 결과를 Local Display에 출력할 수 있게 해 준다. 이때 Remote Machine은 Super-Computer가 될 수 도 있고 또는 Database를 유지하고 있는 임의의 Machine이 될 수 있으며, 이를 실행하기 위해서는 응용 프로그램을 바꿀 필요가 없다. 즉 사용자가 Remote CPU를 이용할 때에도 그 Remote CPU의 Logical Name을 지정하는 것만을 제외하고는 마치 자신의 Local CPU를 사용하는 것과같이 사용할 수있다. X의 진정한 힘은 이러한 Local Area Network 환경으로부터 나온다고 보아도 된다. 이와같이 X-Window의 네트워크 Transparency를 이용하면 Stand-alone에서 부터 Distributed System까지 광범위한 컴퓨팅 스타일이 공존할 수 있으므로 작은 오피스에서는 물론 큰 캠퍼스에도 적절한 컴퓨팅 환경을 쉽게 구축할 수 있다.

2) Vendor 및 기종 독립성 (Independency)

어떠한 하드웨어(Mainframe, WorkStation, PC 등)라도 그위에 X Protocol을 지원한다면, X_based 응용 프로그램들은 그러한 상이한 환경에서 다시 컴파일 되거나, 링크 시킬 필요없이 실행 될 수 있다. X Protocol에 의해 하드웨어의 차이점은 사용자로부터 감추어진다.

3) 정보 공유 기능 (Information-Sharing Capabilities)

한 워크스테이션에서 여러개의 응용프로그램이 실행될 때 응용프로그램 간의 정보교환이 필요한 경우가 있는데 이런경우를 위하여 X 는 다음과 같은 방법을 제공한다.

가) Cut-and-Paste 방식으로 File 간에 간단한 내용을 옮길 수 있다

나) X Properties를 통하여 Structured Data를 공유한다.

Properties는 Window에 첨가될 수 있는 Items으로써 이름, Data Type, Data Format을 가지고 있다.

4) Concurrency

X Window System은 여러가지 응용프로그램이 한 Workstaion 상에서 동시에 실행 될수 있도록 해준다. 보통 각 응용프로그램은 다른 윈도우를 이용하며 각 응용프로그램이 서로 독립적으로 수행되도록 한다.

이 밖에도 마우스 동을 이용하여 이벤트로 수행되는 입력기능과 계층 윈도우를 취급할 수 있는 출력 기능 등 다양한 기능을 갖고 있다.

2. Xlib와 Window Model

가. Xlib Programming

응용 프로그램과 X_based Window System과의 Interface는 오직 X Network Protocol에 의해서만 가능하다. Xlib는 이러한 X Protocol에 Interface 하기위해 만들어진 C language subroutine package로서, Xlib 루틴을 호출하면 X Protocol Request가 생성되어 Display connection을 통해 Server에게로 전송된다. Xlib로 작성된 응용 프로그램은 Server로부터 Event를 받아 그 Event에 대해 원하는 행위를 수행하도록 Server와 interaction한다. 그 주된 기능은 다음과 같다.

1) Server와의 Connection확립 : `XOpenDisplay(...);`

X Window 시스템을 구성하는 Server와 Client 프로그램은 일반적으로 네트워크에 연결되어 여러 Host 상에서 각각 분산 배치된 상태로 돌아 간다. 즉 다른 컴퓨터에서 수행한 것을 display 만 local 컴퓨터에서 할 수 있는 것 이다. 이때 어플리케이션 프로그램이 최초로 해야할 절차는 네트워크 상의 어떤 Server를 사용할 것인가를 결정하여 이 Server와 display 접속을 확립하는 것이다.

2) Window 생성 : `XCreateWindow(...);`

Server로 하여금 Window를 생성하게 한 후, 그 Window의 ID를 반환받는다. 하지만 생성된 직후의 Window는 Unmap 상태이므로 Screen 상에 나타나지 않는다.

3) X Resource 생성 :

Resource란 Server가 관리하는 추상 자료구조체를 말하며, 대표적인 것으로 Window, Graphic Context, Font, ColorMap, PixMap,Cursor 등이 있다. Resource는 Client의 요구에 의해 Server 측에 생성되고 Client는 단지 Resource ID를 받게된다. Server가 이와같이 Resource를 관리해 줌으로써 Client는 통신선을 통해 Resource를 전송할 필요가 없으므로 Network의 load가 극소화 된다.

`XLoadFont(..);` `XCreateGC(..);` 와 같은 Xlib 함수를 사용하여 Resource를 생성한다.

4) Event를 요청 : `XSelectEvent(..);`

Server 측에서 발생하는 다양한 상태 변화, 예를 들어 마우스 키보드의 상태, Window 구성 등이 변했을 때 Event 메시지가 Server로 부터 Client에게 전해진다. 이 Event는 응용 프로그램에서 입력으로 받기를 원하는 Event를 Server에게 알릴 때 사용된다.

5) Window Mapping : `XMapWindow(..);`

Server로 하여금 Window를 화면상에 출력하도록 요청한다.

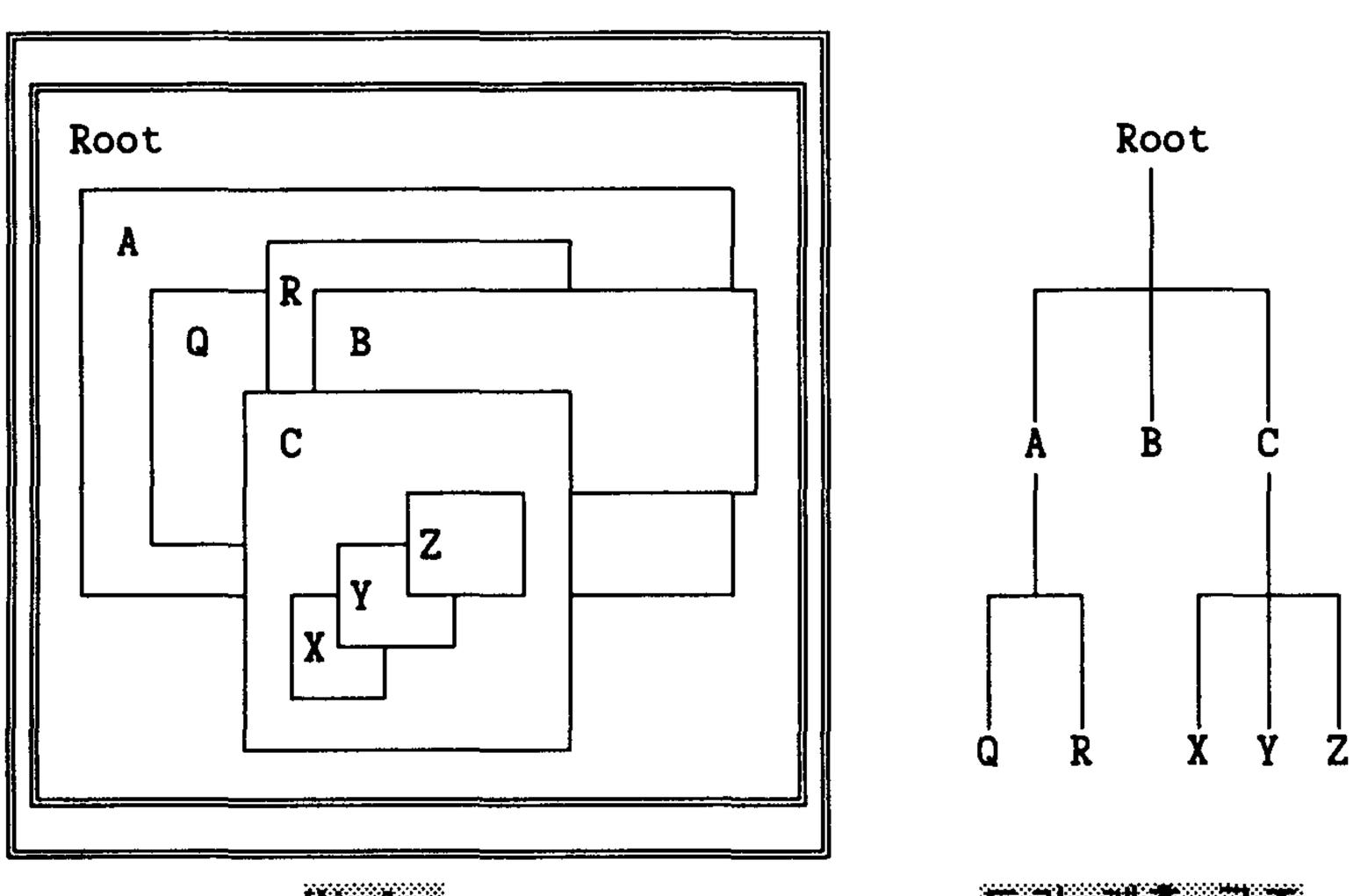
6) Event Dispatching Loop로 진입 :

Event를 처리하기 위한 Event loop은 먼저 Event queue로 부터 Event를 읽어, Event type을 조사하고 Event type 별로 처리하는 loop이다. 일반적으로 while(1) { ...; `XNextEvent();` ... };의 무한 루프에 의해 Event가 처리된다.

나. Window Model

X-Window는 트리형태의 계층구조를 갖는다. 특히 Screen의 전면을 뒤덮는 특별한 Window를 Root Window라 하며, 이것은 트리 계층구조에서 항상 Root로 존재하며, 제거되거나 크기를 변경시킬 수 없다.

응용 프로그램들이 사용하는 Window들은 이 Root Window의 Children이며, 모든 이러한 응용 프로그램들의 각각의 Window들은 Stacking Order를 갖는 Sibling들이 된다. 또한 Screen안의 모든 Window들은 Root Window를 정점으로 하향식의 계층 구조와 left-to-right의 Stacking Order를 갖는 하나의 트리를 형성한다.

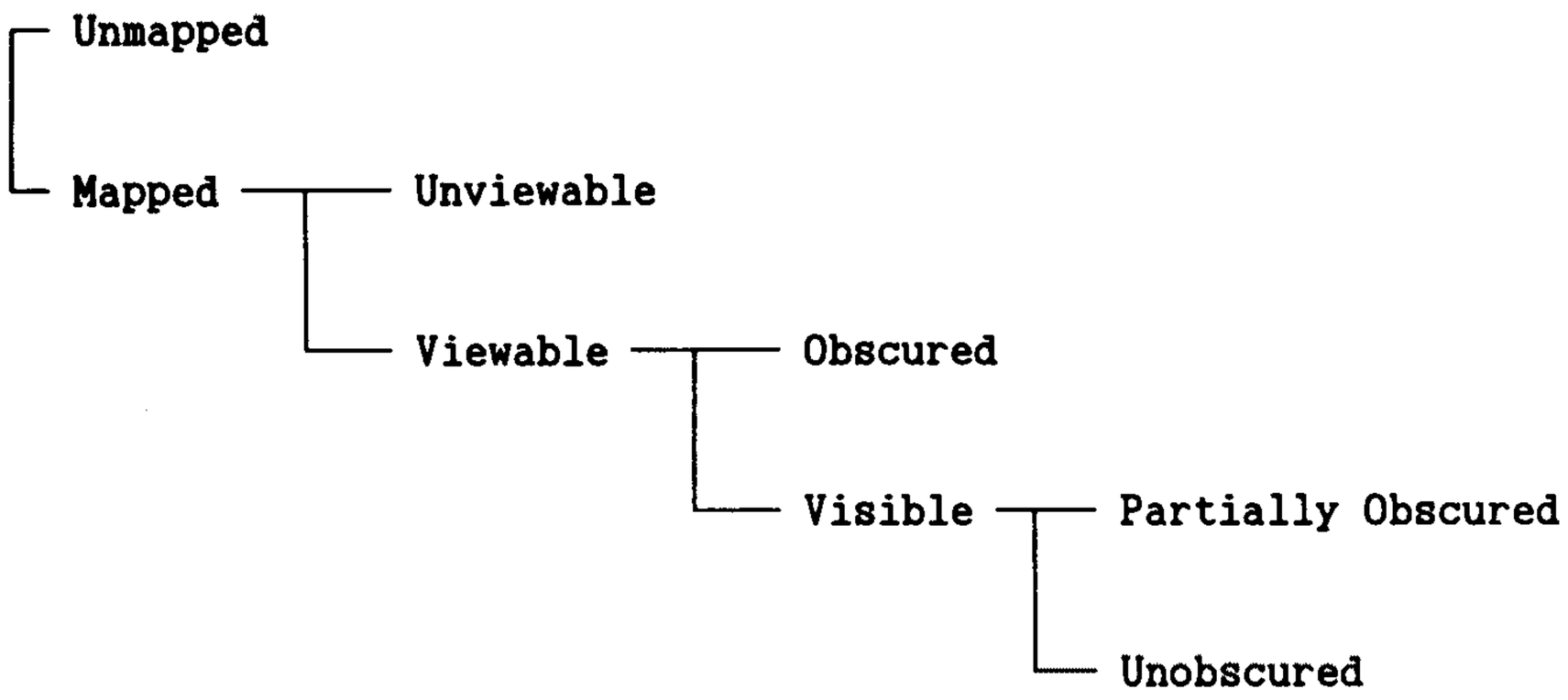


[그림 A-3] Window 계층 구조

Window는 Resource중의 하나이며, 다음 상태중의 하나를 갖는다.

Unmapped	Window를 생성한 후, Mapping을 안한 상태
UnViewable	Mapping을 해서 그 Window는 Mapped상태이나, 그 Window의 Ancestor Window들중의 하나가 아직 Mapping이 안된상태.
(completely) Obscured	모든 Ancestor Window와 해당 Window가 Viewable한 상태이나, 다른 Window에 의해 완전히 가려져 사용자에게는 보이지 않는 상태.
Partially Obscured	모든 Ancestor Window와 해당 Window가 Viewable한 상태이나, 다른 Window에의해 부분적으로 가려져 사용자에게는 Window의 일부분이 보이지 않는 상태
Unobscured	Window의 모든 부분이 사용자에게 보이는 상태.

Window가 Viewable 상태가 되면, 응용 프로그램은 그 Window에 대해서 그래픽이미지를 출력하거나 또는 Server로부터 Event를 받을 수 있다. Window는 다른 응용프로그램에 의해서 언제라도 Unmap될 수 있다. Window Manager는 사용자의 지시를 받아 특정 Window를 Unmap 시키는 기능을 갖고 있다. Window의 Iconifying이 그 예이다.



[그림 A-4] Window 상태 : Mapping, Viewability, Visibility

다. 응용 프로그램 작성 및 실행

응용 프로그램을 작성하여 실행 시키는 절차는 다음과 같다.

1) 소스 프로그램 작성

```
#include <X11/Xlib.h>

#include <X11/Xutil.h>
```

일반적인 C 프로그램에 위와 같은 File을 include 한다.

2) 컴파일과 링크

```
cc -o execution_file_name source_file_name.c -lX11
```

컴파일 하고 X11 Library를 링크 한다.

3) 실행

```
execution_file_name
```

제 3 절 X Toolkit과 OSF/Motif

고 해상도의 Raster 그래픽 화면을 지원하는 Network_based X Window System은 X Protocol을 이용하여 어느 기종의 Unix Machine에서도 일관성 있는 Interface가 가능하지만 Xlib로 응용 프로그램을 작성하는 것은 프로그래머에게는 너무 많은 양의 Coding 이었다. 이 문제를 해결하기 위하여 응용 프로그래머들은 User Interaction과 High-Level의 그래픽 라이브러리가 새로운 시스템의 기저로 되어야 한다고 생각하게 되었고, 이를 뒷받침 할 수 있도록 Development Toolkit을 개발하는 Project 들이 시작되었다. 그러한 Project중의 하나가, MIT, DEC 그리고 HP등이 참여한 X Toolkit Project이다. X Toolkit은 User Interface를 쉽게하기위한 Xlib 위에 있는 Application Interface Layer로서 Widget Set과 Intrinsics 구성되어 있고 Widget은 X Toolkit에서 User Interface Construction을 위한 기본적인 Entity로 각각 속성과 모양을 갖고 있다. Widget Writer의 관점에서는 Widget은, 한개의 X Window와 그 Window에 대한 Widget-specific Action을 수행하는 방법(a set of procedure)이 결합된 한개의 User Interface Object(복합자료구조)로 볼 수 있다. Xt (X Toolkit) Intrinsics는 Xlib call로 조합된 상위 level의 library call로서 부품 (메뉴나 Push Button 등 Widget)을 이용하여 User Interface를 실현 시킨다. 실제로는 이런 부품은 Xt call 로도 작성 가능하지만, 프로

그래머의 편의를 위해 OSF/Motif Widget set 같은 Xt call을 간략화한 상위 level의 library가 개발되었다. OSF/Motif 외에도 MIT의 Athena Widget Set, AT & T 의 OPEN LOOK Widget Set 등이 그 예 이다.

1. Xt Intrinsics

가. X Toolkit Programming

기존의 Xlib로 짠 프로그램은 Event-driven 모델로 되어 있지만 X Toolkit 프로그램은 프로그래머의 logic 부담을 줄여주기 위한 Dispatch-driven 모델이다. 기존의 Event-driven 모델에서는 각 Event의 타입을 case문에 써서 Action을 취해주어야 하나, Dispatch-driven 모델에서는 각 이벤트를 미리 등록해 놓았다가 이벤트 발생시 해당 유저 인터페이스 컴포넌트에게 dispatch해주도록 되어 있다. 그러므로 프로그램이 선언문 형식 (A 발생시 B를 하라) 으로 되어 있어 프로그램 하기가 쉽다. X Toolkit Programming Model에 의해 Xt Intrinsics를 사용하여 프로그램을 작성할 때, 그 기본구조는 응용 분야에 관계없이 다음과 같은 절차를 갖게 된다.

1) Intrinsic을 초기화함 : `XtInitialize(..);`

X Server에 대한 connection 확립과 Resource를 할당하고 Intrinsic Layer를 초기화 한다.

2) Widget을 생성 : `XtCreateWidget(...);`

프로그램의 User Interface를 위해 필요한 Widget을 생성한다.

3) Callback Function과 Event Handler를 등록 :

`XtAddCallback(...);` `XtAddEventHandler(...);`

각 Widget 내에서 발생하는 User Action과 Event에 대해

반응하는 사용자 정의 함수를 등록한다.

4) Widget들을 realize한다 : `XtRealizeWidget(...);`

Widget 들을 실체화(realize) 함으로써 X Window를 생성시킨다.

5) Event Loop에 진입 : `XtMainLoop(...);`

대부분의 X 응용 프로그램은 Event-driven방식이다.

무한 Event 루프에 들어가 X Envent Queue에 있는 Event를 꺼내

와서 그 Event에 관련된 Event Handler나, 그 Event가 발생한

Widget과 결합된 Callback Function을 수행하도록 한다.

나. Xt Intrinsics의 기본 함수

1) Initialization

Client와 Server간의 연결을 확립하기 위하여 초기화 하는 Function

인 XtInitialize(name,class,options,noptions,&argc,argv)를 쓰면 X

자원 매니저에 의해 사용되는 자원 데이타베이스를 초기화 한다.

이 Function은 TopLevelShell Widget을 생성하여 모든 다른 Widget의

Container로서 쓰인다. XtInitialize는 다음과 같은 일을 한다.

가) Parameter에 의해 Resource file 과 Command_line Option으로 지정된 Resource 값을 읽어 드린다.

나) Display의 Parameter에 의한 환경변수에 따라 XOpenDisplay()로 적절한 Server에 연결한다

다) 읽어 들인 Resource를 기본으로 Application내의 Toplevel Widget과 Shell Widget을 작성한다. Shell Widget이 작성되면 그 Shell Widget의 Window ID가 생기는데 이는 실제로는 Widget 이름_widgetRec 구조체에의 Pointer가 생기는 것이다. Application Program에서 User가 마우스로 Toplevel Widget의 Size를 변환시키려면 그 정보는 Window Manager --> Shell Widget --> Composite Widget으로 전해진다.

2) Widget의 생성

Root Widget은 Shell Widget이라고 하며 오직 하나의 Child Widget만 갖게 되어 있지만 다른 Widget 들은 여러개의 Widget을 Child로 가질 수 있다. XtInitialize()에 의해 만들어진 Toplevel Widget을 Parent로서 그의 Child Widget, Grandchild Widget을 차례로 만든다. Toplevel의 child는 1개 밖에 쓸 수 없으므로 Toplevel 밑에 Form

이나 RowColumn 등의 Container Widget을 배치한다. Widget을 작성하려면 X Toolkit call을 직접쓰는 방법과 Motif call로 작성하는 방법이 있다.

다음 [표 A-1] 에 Class명, Widget을 작성하기 위한 Motif call 과 Header File이 수록 되어있다.

XtCreateWidget (name,class,parent,args,nargs)

예) XtCreateWidget ("scroller",xmScrollBarWidgetClass,parent,
NULL,0);

Motif Widget에 대한 Header File은 /usr/include/Xm에 있다.

만일 프로그램이 Widget Resource 들을 지정할 필요가 없는 경우에는 args는 NULL이 된다.

[표 A-1] Class 명, Widget을 작성하기 위한 Motif call 과 header file

Class 명	XmCreate xxxx()	Header file
XmArrowButton	XmCreateArrowButton()	<ArrowB.h>
XmArrowButtonGadget	XmCreateArrowButtonGadget()	<ArrowBG.h>
XmBulletinBoard	XmCreateBulletinboard()	<BulletinB.h>
XmCascadeButton	XmCreateCascadeButton()	<CascadeB.h>
XmCascadeButtonGadget	XmCreateCascadeButtonGadget()	<CascadeBG.h>
XmCommand	XmCreateCommand()	<Command.h>
XmDialogShell	XmCreateDialogShell()	<DialogS.h>
XmDrawingArea	XmCreateDrawingArea()	<DrawingA.h>
XmDrawnButton	XmCreateDrawnButton()	<DrawnB.h>
XmFileSelectionBox	XmCreateFileSelectionBox()	<FileSB.h>
XmForm	XmCreateForm()	<Form.h>
XmFrame	XmCreateFrame()	<Frame.h>
XmLabel	XmCreateLabel()	<Label.h>
XmList	XmCreateList()	<List.h>
XmMainWindow	XmCreateMainWindow()	<MainW.h>
XmMenuShell	XmCreateMenuShell()	<MenuShell.h>
XmMessageBox	XmCreateMessageBox()	<MessageB.h>
XmPanedWindow	XmCreatePanedWindow()	<PanedW.h>
XmPushButton	XmCreatePushButton()	<PushB.h>
XmPushButtonGadget	XmCreatePushButtonGadget()	<PushBG.h>
XmRowColumn	XmCreateRowColumn()	<RowColumn.h>
XmScale	XmCreateScale()	<Scale.h>
XmScrollBar	XmCreateScrollBar()	<ScrolledBar.h>
XmScrolledWindow	XmCreateScrolledWindow()	<ScrolledW.h>
XmSelectionBox	XmCreateSelectionBox()	<SelectionB.h>
XmSeparator	XmCreateSeparator()	<Separator.h>
XmSeparatorGadget	XmCreateSeparatorGadget()	<SeparatoG.h>
XmText	XmCreateText()	<Text.h>
XmToggleButton	XmCreateToggleButton()	<ToggleB.h>
XmToggleButtonGadget	XmCreateToggleButtonGadget()	<ToggleBG.h>

3) Managing Widgets

여기에서 Widget의 manage란 Parent Widget이 그 Child Widget을 조사하여 어떻게 배치할 것인가를 결정 (Geometry Management)하는 것이다. 어느 Widget을 manage 하는 것은 그 Widget을 Parent Widget에 의한 geometry management의 대상 list에 첨가하는 것을 의미한다. 이것을 XtRealizeWidget call을 하면 Widget 용의 Window가 처음으로 생기게 된다. 그 후 Application 프로그램이 XtMainLoop()을 call 하여 Event를 기다리는 무한 loop에 들어가면 Window가 나타난다. Widget의 manage와 map의 차이는 unmap 에서는 그냥 보이지만 않을 뿐 장소는 차지하고 있지만 unmanage에서는 지정된 Widget을 Parent의 Geometry Managent 대상에서 제외시키기 때문에 장소도 차지하지 않으며 Widget들이 재배치된다.

Shell Widget을 제외한 모든 Widget은 그 부모 Widget에 의해 관리된다. 부모 Widget은 자녀 Widget의 크기와 위치를 관리하고, 자녀 Widget의 map 여부를 결정하며, 입력 focus를 조종하여 자녀 Widget의 입력도 제어한다.

XtManageChild(Widget)

XtCreateManagedWidget(name,class,parent,args,nargs)

4) 이벤트 Dispatching

Xt Intrinsics는 이벤트가 발생한 윈도우에 해당하는 Widget을 살펴보고 이벤트 핸들러가 있으면 이를 호출한다. 이와같이 해당 Widget을 살펴본 다음 이벤트를 호출하는 것을 Dispatching이라 한다.

XtMainLoop() : Event loop로서 그 코드는 다음과 같다.

```
while(TRUE)  {
    XEvent event;
    XtNextEvent (&event);
    XtDispatchEvent(&event);
    }
```

5) Setting Widget Resources

프로그램을 실행시키기 전에 Resource File 을 준비한다. Resource File이란 여러가지의 Widget 속성(위치, label문자열 등)을 지정하는 것이다. 간단히 Resource File을 설정하기 위해서는 Resource File에 지정해 주거나 다음과 같이 프로그램 내에서 Resource를 지정할 수가 있다.

```
typedef struct {
    String name;
    XtArgVal value;
} Arg, *ArgList;
```

```
static Arg wargs[] = {
    { XtNwidth, 300},
    { XtNheight, 400},
};

XtCreateWidget("sample",xmPushButtonWidgetClass,
                parent, wargs, XtNumber(wargs));
```

또는

```
Arg args[10];
int n = 0;
XtSetArg(args[n], XtNwidth, 300); n++;
XtSetArg(args[n], XtNheight, 400); n++;
XtCreateWidget("sample", xmPushButtonWidgetClass,
                parent, args, n);
XtSetValues(widget, arglist, nargs)
```

다. 이벤트 처리 함수 등록

각 Widget에서 사용자의 Action이 발생하면, 이것을 Server에게 이벤트로서 넘겨준다. 그러면 이 이벤트 타입을 위한 이벤트 핸들러나 Callback 리스트에 등록된 함수들(Callback 이라고 하며 프로그래머가 등록함)이 호출됨으로써 이벤트가 처리된다. 이벤트처리 함수를 등록하는 방법은 다음과 같다.

1) 이벤트 핸들러에 등록

한 Widget안에서 특정 Event가 발생했을때 Intrinsic에 의해 불리워지는 함수를 Event Handler라 하며, 이러한 Event 발생에 대해서 프로그래머는 Event Handler를 정의한 후, X Toolkit 루틴인 XtAddEventHandler(...); 를 이용하여 등록 할 수 있다.

단 XtAddEventHandler()의 Event_mask 매개변수는 화일 <X11/X.h>에 정의된 표준 Event mask가 되어야 한다. (NoEventMask, KeyPressMask, KeyReleaseMask, ButtonPressMask, EnterWindowMask, PointerMotion Mask, ExposureMask, OwnerGrabButtonMask 등 26개의 마스크가 <X.h>에 정의 되어있다.)

```
XtAddEventHandler(widget,eventmask,nonmask,handler,client_data);
```

Eventmask : 이벤트 타입으로서, 화일 X,h에 등록된 표준 Event mask

nonmask : FALSE=>이벤트 타입 지정

TRUE =>이벤트 타입 지정하지 않는 경우

handler : 이벤트 핸들러명

client_data : 이벤트 핸들러에게 패스되는 아규먼트(프로그램 지정)

or NULL

```
예)XtAddEventHandler(msg_widget,ButtonPressmask,FALSE,quit,NULL);
```

2) Callback 함수의 등록

각 Widget은 제각기 그 Widget에 따라 정의된 Callback list를 갖고

있다. Callback온 프로그램 실행시 그 Callback condition을 만족시킬 때 프로그래머가 정의한 함수를 call할 수 있도록 만들어진 Mechanism 이다. Callback list는 Widget에 따라 그 entry가 다르므로, 사용시 각각의 Widget에 대한 Documentaion을 참조해야 하며, Callback Function의 등록을 위해서는 X Toolkit 루틴인 XtAddCallBack(...);을 사용한다. Callback Function은 특정 Widget 의 특정 Callback Resource에 등록이 되기때문에 같은 Widget Class에 속하는 2개의 Widget은 같은 Callback Resource에 대해 서로 다른 Function을 정의 할 수 있다. 이는 이벤트의 처리를 어떤 특정한 이벤트에 구속 시키지 않고, 좀 더 abstract한 유저의 Action에 따라 수행할 수 있도록 해 준다. Callback 리스트에 Callback 함수를 등록시키는 방법은 다음 과 같다.

```
XtAddCallback(widget,callback_name,proc,client_data);
     callback_name: callback 리스트명
     proc:          callback 함수명
예) XtAddCalback(msg_widget,XmNactiveCallback,quit,Null);
```

3) Translation Manager의 이용

Translation Manager는 사용자의 Action을 Widget 또는 프로그램에서 제공되는 함수(Action Procedure)로 Mapping 하는 도구이다.

Action은 Resource 화일(Event-Action pair)과 Resource Manager에 의해 관리되거나, X Toolkit 루틴인 XtAddAction(..)에 의해 프로그램 코드상에서 정의 될수 있다. Callback의 경우 Widget Writer에 의해 미리 User의 Action이 예상된 경우에, Callback Resource를 Widget에 첨가시킨 것이고, 반면에 Action은 응용프로그래머나 사용자가 임의의 특정 Event(combination of mouse or key stroke)에 원하는 Function을 결합시킬 수 있도록 하기위해 제공되는 것이다. 또한 Widget내의 Method의 Binding이 프로그래머 또는 사용자의 기호에 맞지 않는 경우 이 Action을 이용하여 변경할 수 있고, 사용하는 Widget에서 원하는 기능의 Method를 제공하지 않는 경우 Action을 이용해 Widget의 특정 Event에 대해 새로운 Method를 정의함으로써 Widget의 행위를 확장 시킬수 도 있다. Action을 등록하는 방법은 다음과 같다.

가) 사용자의 Xdefaults 화일에 지정

예) memo*XmPushButton.translations: <Key>q: ArmAndActivate()

memo: program명

*: wild character

XmPushButton: widget명

이 문장은 사용자의 Action "<Key> q"와 Action 프로시듀어 ArmAndActivate()를 결속시켜 준다. Action Procedure ArmAndActivate()는 XmPushButton Widget에 의해 정의되며, XmNarmCallback 리스트

와 XmNactivateCallback 리스트 내에 존재하는 Callback 함수를 호출한다.

나) 함수를 이용한 등록

o Translation의 정의

예) static char defaultTranslations[] = "<Key>q: bye()"

o Action 프로시듀어의 등록

형식: XtAddActions(actions,num_actions)

o Translation table의 형성

형식: XtParseTranslationTable(source)

예) trans_table=XtParseTranslationTable(defaultTranslations);

o Translation table의 등록

형식: XtAugmentTranslations(widget,translation_table) 또는
XtOverrideTranslations(mest_widget,trans_table);

예) XtAugmentTranslations(msg_widget,trans_table);

라. Resource Setting

Widget으로 User Interface를 작성할때 주요한 설계원칙중의 하나는, User Interface의 가능한한 많은 부분들이, 응용프로그램의 End-user에 의해 customize되도록 허용하는 것이었다. 이를 위해 X Tool kit은, 실행시 Configuration을 Xlib Resource Manager 에 의해 구현되도록 하고있다. 응용 프로그램에서 사용되는 모든 Resource의 속성들을 프로그램 코드상에서의 Hard-coding이 아닌, Resource 화일로

분리할 수 있도록 허용함으로써, Source코드에는 허드렛 일들을 하는 코드의 부분을 줄임으로써 Readability와 프로그램 전체 구조를 단순화시키는데 도움을 준다. 또한 응용 프로그램을 시스템에 장착할 때, 사용하는 시스템에 가장 잘 맞는 색상과 폰트를 고르는 경우 Source Code를 건드리지 않고 단지 Resource 화일로 End-user의 기호에 맞게 조정하는 것이 가능하여, 결국은 응용 프로그램의 유연성 증가 시키는 역활을 해 준다.

Resource의 대표적인 예로 Font, Foreground Color, Background Color, BorderWidth, LabelString, X, Y, Width, Height, Colormap등이 있다. 이들 Resource는 각 Widget에따라 정의되어 있으므로, 사용시에는 해당하는 Widget의 Documentation을 참조해야 한다. 한 Widget의 Resource들은 그 Widget의 상위 Class에 있는 Widget의 Resource를 모두 상속 받으므로, 그 Widget의 완전한 Resource를 보려면 상위 클래스의 Resource도 함께 보아야한다. Widget의 Resource 속성값 지정이외에, Resource Manager는 Translation Management 기능에 의해 키보드의 키(또는 마우스의 button)와 Widget의 action routine을 결합시킬 수 있도록 해 주고 있다. Tarnslation Management기능에 의해 Server로부터 넘어오는 Event를 Client 프로그램의 Action과 연결시켜 줌으로써, 응용 프로그램상에서 Hot-Key와 같은

기능을 쉽게 구현할 수 있다.

1) Resource File

프로그램을 실행시킬때 Resource File을 이용하는데 Resource File은 Program 내의 각 Widget의 Resource값을 지정하는 File이다.

Resource란 사용자가 임의대로 변경하여 이용할 수 있는 Data로서 윈도우 ID, Color, Font, Image, Text, Widget명,윈도우의 위치나 크기 등 프로그램에 영향을 줄 수 있는 모든 것 들을 말한다. 이러한 자원의 값은 프로그램을 실행시킬때 Command-line option 으로 명시하거나, 미리 특정 화일에 그 값을 설정해 둘 수도 있다. 아무 곳에도 명시되지 않은 자원들의 값은 시스템이 Default로 설정한 값을 이용하게 된다. 각 Widget의 Resource값은 프로그램내에 지정되어 작성하는 경우와 Resource File에 기술하는 경우 2가지가 있다.

2) Resource File 지정 방법

모든 자원은 자원의 이름과 클래스명을 갖는다. 자원값 지정시는 자원명과 클래스명을 사용하여 지정 가능하다.

가) 자원명으로 지정

공통으로 사용되는 자원명은 StringDefs.h 화일에 지정되어 있다. 자원명은 소문자로 시작하며, 그 지정 방법은 다음과 같다.

예) draw.panel.commands.button1.foreground:red

draw*button1.foreground: red

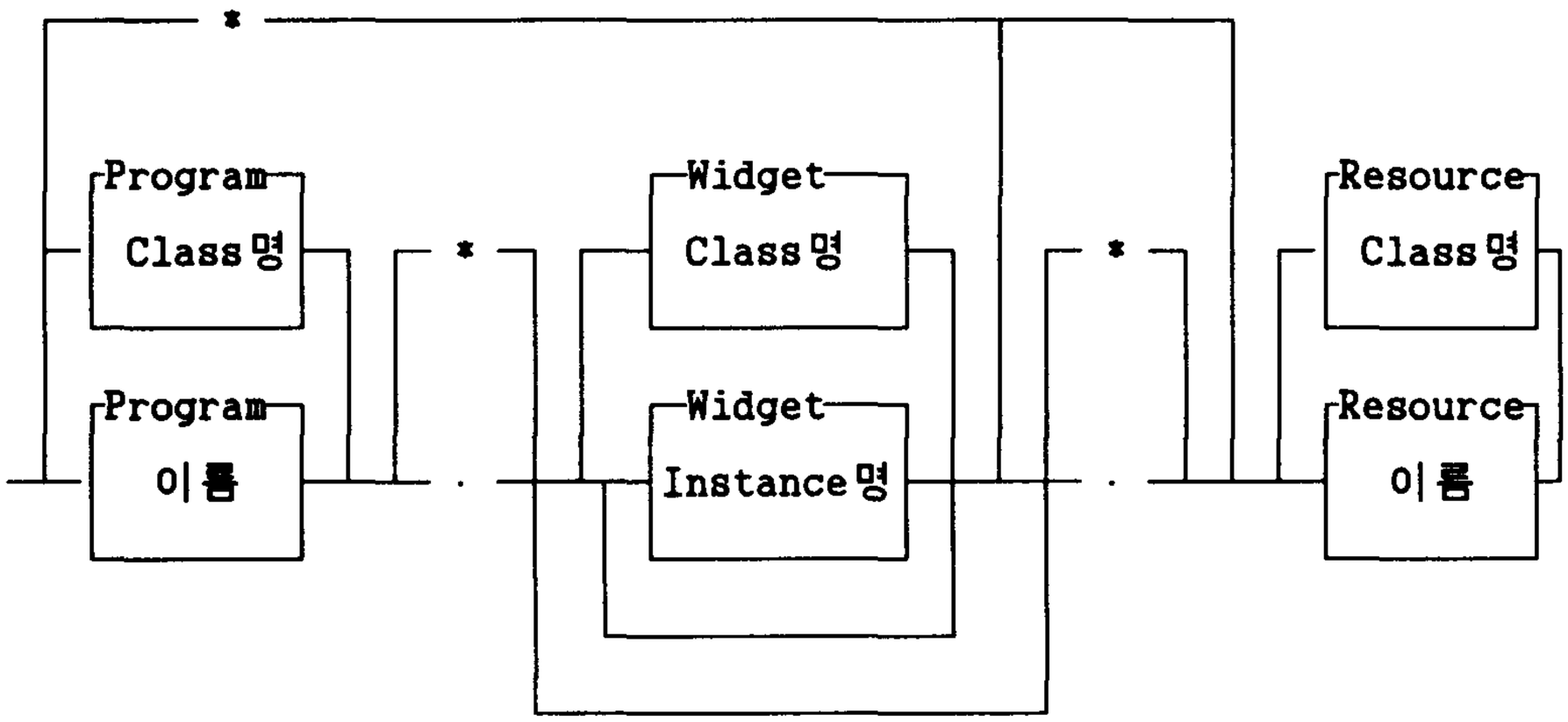
나) 클래스명으로 지정

Class별로 자원을 지정하며 자원명의 경우보다 범위가 넓다.

클래스명은 대문자로 시작하며 그 지정방법은 다음과 같다.

예) Draw.XmBulletinBoard.XmRowColumn.XmPushButton.Foreground:red

*.XmPushbutton.foreground: green



[그림 A-5] Resource File 서식

예) Sample.bb.pbl.foreground : red

여기에서 XmNforeground는 XmPushButton의 Super-Class인 XmPrimitive의 Resource이고, XmPushbutton의 Resource로서도 사용될 수 있다. Resource명은 보통 XmN이고, Resource Class명 XmC로 시작하는 문자열로 되어 있지만 Resource File에는 XmN, XmC는 생략된다. Program명과 Program Class명은 XtInitialize ( )의 제1, 제 2 Parameter로 지정된 문자열이다. 만일 수행시킬때 Option을 주는 경우 % sample -n other 라고 하면 Resource File에서 Sample로 시작하는 Resource는 무시하고 other로 시작하는 other *bb.maginWidth : 50등이 이용된다.

예를 들어 bb/pb1/pb2의 XmNforeground의 값을 red로 지정하고 싶은

경우에는 Sample.bb *foreground : red 라고 하면 된다.

Resource File에서 .과 *의 차이는 다음과 같다.

.는 Tight binding	.의 우측과 좌측은 Parent.Child관계이어야 함
*는 Loose binding	*의 좌측의 Widget는 우측 Widget의 직접적인 Parent일 필요는 없고 Ancestor이면 된다.

3) Resource 기술의 우선 순위

Resource File내에서 복수의 Resource지정이 있는 경우 아래 규칙에 의해 Resource값이 채용된다.

가) Resource File에서 후에 지정된 것.

나) 보다 상위 Widget Instance에 대한 기술이 있는 쪽이 우선.

예) Sample * bb.pb1.labelString : WIN (쓰임)
   Sample *pb1.labelString : NOP

다) Class 명에의 지정보다 Instance명이 지정된 쪽이 우선.

예) Sample *pb1.labelString : WIN (쓰임)
   Sample *XmPushButton.labelString : NOP

라) (*)으로 결합된 것보다 (.)로 결합된 쪽이 우선.

예) Sample *bb*pb1.labelString : WIN (쓰임)
   Sample *pb1 *labelStrin : NOP

마) 이름이나 Class명이 명시적으로 지정된 쪽이 우선

예) Sample *pb1.labelString : WIN (쓰임)
   Sample *labelString : NOP

4) Resource 데이타베이스를 Load하는 순서

XtInitialize() 는 자원 데이타베이스를 생성하여, 사용자 자원화일을 load시키고, Command-line option(아규먼트 argv 와 Option)에서 지정한 자원을 추가시킨다. 프로그램 상에서 지정한 자원의 값이 있을 경우(아규먼트 args로 지정)는 이것이 최우선 순위이다.

자세한 load 순서는 다음과 같다.

가) Load /usr/lib/X11/$LANGapp-defaults/<class>.

나) If step 가) fails, load /usr/lib/X11/app-defaults/<class>.

다) Load $XAPPRESLANGPATH<class>.

라) If 다) fails, load $XAPPLRESDIR.

마) Load data in RESOURCE_MANAGER property.

바) If 마) fails, load $HOME/.Xdefaults.

사) Load File specified by $XENVIRONMENT.

아) If 사) not set, load $HOME/.Xdefaults<host>.

자) Load command-line options.

Widget이 생성될 때 자동적으로 자원을 retrieve 한다. 그러나 자원 매니저를 이용하여 프로그램 단계에서의 옵션이나 자원을 retrieve 할 수 있다. 자원 데이타베이스로 부터 자원을 retrieve 하기 위해서는 /usr/include/X11/Xresource.h 에 정의된 XtResource 구조를 이용한다.

2. OSF/Motif

OSF/Motif Widget이란, OSF(Open Software Foundation - DEC, HP, IBM 등의 Vendor로 구성된 조직으로 Operating System의 표준화를 위해 조직됨)가 제창하고 있는 X Window상에 상위 Level의 GUI(Graphical User Interface)를 제공하는 Object 이다. 즉 User Interface에 필요한 Popup Menu, Push Button, Scale, List등을 작성하기 위한 Widget library call을 한다. OSF/Motif Widget Set은 Xt call을 간략화한 Widget Level의 Motif library call을 이용하고 있다. Widget set에는 OSF/Motif Widget 외에도 MIT 의 Athena Widget Set, AT&T의 Open Look Widget Set등이 있다.

가. Motif Widget의 특징

Motif Widget에는 다음과 같은 특징이 있다.

1) X Widget의 기본 성질인 Network 지향적이고 X Protocol을 이용하며, Client Server 모델에 기반을 두고, Divice, OS, Vendor 독립성을 지향하고 있다.

2) OPEN LOOK과 함께 업계의 표준이다.

3) Widget은 각각이 Window이고, Window처럼 Parent, Child Window가 있다.

4) Geometry Management 기능이 있어서 User Program은 각각의 부품

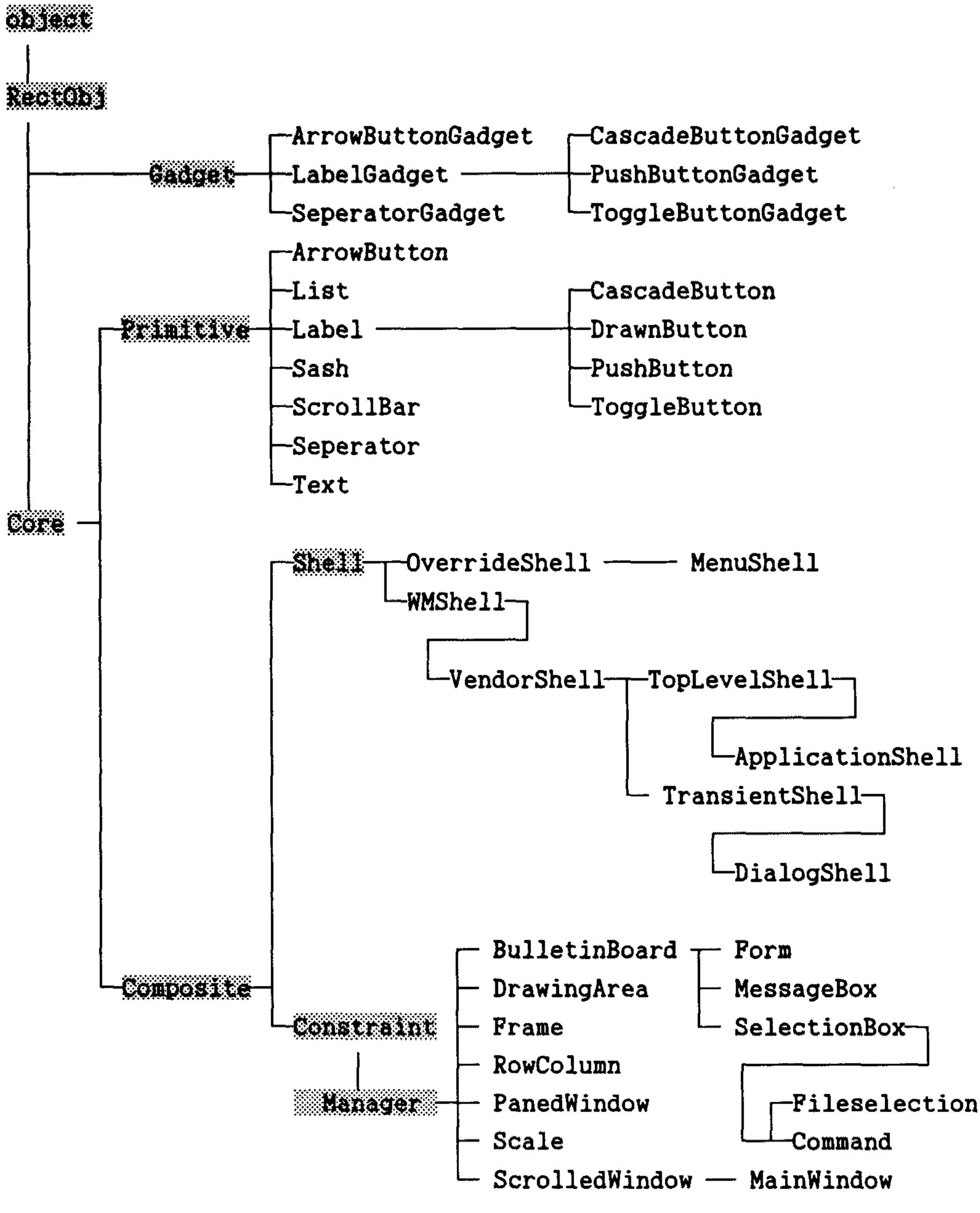
의 크기를 지정할 필요가 거의 없다.

5) Object Oriented Programming을 기반으로 설계되어 있다.

X Toolkit Intrinsic은 User Interface Component라 불 수 있는 Widget들을 결합하여 완전한 User Interface를 구성하도록 해주는 골격이다. 반면 Widget은 하나의 X Window와 그 Window에 관한 Widget-specific method들과 Resource와 같은 정보들을 저장할 수 있는 자료구조를 갖고 있는 하나의 Object로 볼 수 있다. 현재 만들어져 나온 대표적인 Widget set은 다음과 같다.

- X Widget set (Hewlett Packard)
- Xsw Widget set (Sony)
- Athena Widget set (MIT's Athena project)
- Cornell Widget set
- Motif Widget set (OSF : Open Software Foundation)
- XT+ toolkit (AT&T : based on the OPEN LOOK Interface style)

나. Motif Widget Set 계층 구조



▒ : X Toolkit Intrinsic으로 정의된 Class로 모든 Widget set에 동일.

[그림 A-6] Motif Class Hierarchy

다. Motif Widget Class

1) Widget Class와 계층 구조

Widget library는 object oriented programming style로 되어 있고 Class의 승계 개념이 있는데 Appliation Program을 작성할 때 이것을 염두에 두어야 한다. Widget Library에서는 User Interface를 작성할 때 사용하는 부품을 Widget Class라고 부르는 Group으로 나눈다.

Class는 말하자면, 동일한 성질을 지닌 부품의 집합이라고 할 수 있다. 각 Class에는 그 Class의 부품을 표현하기 위해 공통적으로 사용되는 속성을 정의하고 있다. 이 속성을 Resource라 부르고 있다.

Resource는 속성의 종류에 의해 각각이 명칭을 가지고 있다.

예를 들어 XmLabel Class의 표시문자 에는 XmNlabelString, Font는 XmNfontList라는 Resource명이 정의되어 있다. Class라는 집합의 1개 요소를 Instance라고 부른다. Widget을 사용하는 User Interface의 작성은 Widget을 조합하는 작업이라고 할 수 있다.

2) Class 계승

Widget Class는 그림 [A-6]과 같은 계층 구조로 되어 있다. 계층 구조 중 상위 Class를 Super_Class, 하위 Class를 Sub_Class라고 부른다. Sub_Class는 Super_Class의 기능.성질을 계승하고 있다. 예를들면, Push Button(XmPushButton) Class는 XmLabel Class의 Sub_Class

로 정의되어 있다. 결국 XmPushButton은 XmLabel Class의 기능이나 성질을 물려받고 있다(inherited). 또한 Super_Class의 Resource도 Sub_Class의 Resource로 계승된다. 결국 Push button Widget을 작성하는 경우에는 XmLabel의 Resource인 XmNlabelString의 값을 지정할 수 있다.

3) Meta Class

그림 [A-6]에서 Box로 둘러싸여 있는 Meta_Class는 다른 Class를 정의하기 위한 추상적 Widget Class 이다. OSF/Motif에는 Meta_Class에 다음과 같은 것이 있다.

가) Core Class

Core Widget은 모든 Widget의 Super_Class로의 역할을 하며, 위치 크기, 배경색과 같은 모든 Widget에게 공통적으로 필요한 Resource 와 Callback list를 정의하고 있다.

나) XmPrimitive Class

Push Button이나 Label등의 비교적 단순하고 기본적인(primitive) 부품의 Super_Class이다. Button 들레 형상.색등 공통으로 사용할 수 있는 Resource를 정의하고 있다.

다) Composite/Constraint/XmMangager Class

Child Widget들의 목록을 관리하며, Child의 첨가, 제거, 그리고

Child로부터의 새로운 Geometry에 관한 Request를 관리하고,

Children에게로의 입력 Event할당을 관리하는 Method를 갖고있다.

Constraint Class는 그 Child Widget이 Parent Widget에 대해서

배치될 수 있도록 되어 있다. Constraint Class의 Child Widget은

Constraint Class가 정한 Constraint Resource라고 하는 특수한

Resource를 사용한다. Manager Widget은 Meta_Class이고

그 Resource는 Sub_Class에서 이용된다.

라) Shell Class

Window Manager(mwm등)의 Application Program에는 필수적인 Wid-

get 계층의 Top으로서 Shell Widget이 존재하지 않으면 안된다.

또한 Popup Menu등에는 Override Shell이나 Transient Shell이 반

드시 존재한다. 마우스에 의한 Application Window의 재배치 Size

의 변환등의 Window Manager 요구를 Application 내의 Window

(widget)에 전하는 역할을 한다.

마) Xmgadget Class

Motif에서는 Widget이외에 User Interface Component로 Gadget을

제공하고 있다. Gadget은 그들 자신의 Window를 갖지 못하고, 부

모 Window의 입력에 의존하며, 부모 Window에 Text나 Graphic을

출력 할 수 있다는 점과, Translation과 Event Handler와 Popup

Children을 지원하지 못한다는 점만 제외하고는 대응하는 Widget과 동일하며 Callback Function은 지원한다. 응용 프로그램에 있어서 Window 수를 줄이는 것은 Server Request를 줄이는 것을 의미하기 때문에 Gadget을 사용함으로써 응용 프로그램의 효율을 높일 수 있다. Gadget은 그하나 하나의 Instance가 Window로서 실체 갖지않기 때문에 실행속도가 퍽 개선된 것이다. Gadget에서 사용할 수 있는 부품에는 다음 6가지가 있다.

XmArrowButton Gadget　　XmCarcadeButton Gadget

XmLabel Gadget　　XmPushButton Gadget

XmSeparator Gadget　　XmToggleButton Gadget

4) Motif Widget Class

Widget은 그 기능상 몇 개의 클래스로 나뉜다. 각 클래스는 그 클래스에 속하는 모든 Widget과 관련이 있는 Procedure와 데이타로 이루어져 있다. 물리적으로 말하면 Widget 클래스는 이들 데이타 구조에 대한 포인터 이다. 그리고 이들 Procedure와 데이타는 하위 클래스가 물려 받을 수 있다. 이와같이 각각의 Widget은 고유의 기능을 가지고 있으므로, 프로그램에서 사용시는 그것의 기능을 보고 필요한 Widget 즉 Tool을 선택하여 쓰면된다. 단, 모든 프로그램은 ApplicationShell을 Top-level로 시작하여야 하며, 추가의 Top-level Shell이 필요한

경우는 TopLevelShell을 이용한다. 2개 이상의 Widget을 혼용하고자 하는 경우는 반드시 Container 클래스의 Widget을 그들의 부모로 두고 그 밑에 필요한 기능에 따라 이들 Primitive 클래스의 Widget을 두어야 한다. 위에서 설명한 Core, Composite, Constraint, Xhell, Primitive, Manager Widget Class 등 Meta Class의 하위 Widget Class로서 다음과 같은 Widget Class 들이 있다.

가) TopLevelShell Widget Class :

Shell Widget의 Sub-Class로, Icon Representation에 관련된 Method를 포함하고 있다. 대부분의 응용 프로그램들은 TopLevel Shell Class를 Top Level Widget으로 사용한다.

X Toolkit Intrinsic 루틴인 XtInitialize(...)는 TopLevel Shell을 생성한다.

나) OverrideShell Widget Class :

Window Manager의 개입을 막는다는 점 이외에는 TopLevelShell과 유사하다. OverrideShell Widget은 Override_redirect attribute를 set하여 Window Manager가 Window를 관리하지 못하도록 하며 Popup Menu에 자주 쓰인다.

다) Label Widget Class :

Non-selectable Text나 Menu Title과 같은 문자열 또는 Pixmap을 출력하기위해 사용된다.

라) PushButton Widget Class :

사용자의 입력을 받으면 Button으로 반전된다. GUI의 Point-and-

click Interface를 위한 주요한 Building Block이다.

마) ToggleButton Widget Class :

Binary 상태를 유지하는 버튼이다. On-Off 옵션에 쓰인다.

바) List Widget Class :

항목들중에서 선택하는데 사용된다.

사) ScrollBar Widget Class :

수평, 수직 ScrollBar를 생성하기위해 사용된다.

아) Text Widget Class :

문서 편집 기능을 갖고 있다.

자) RowColumn Widget Class :

자신의 Child 들을 가로/세로로 정렬하는 데 사용된다.

차) BulletinBoard Widget Class :

자신의 Child 들을 Widget내에서 절대좌표 (x,y)에 놓도록 하는 데 사용된다.

카) Form Widget Class :

이 Widget의 Child들의 위치를 Reference Widget의 상대위치에 놓도록 관리한다. Resize시 Widget들의 상대적인 Geometry가 일정하도록 하는데 사용된다.

타) RadioBox Widget :

이 Widget의 Classs는 RowColumn Widget Class에 속한다.

Default로 ToggleButtonGadget을 Child로 하도록 되어있다.

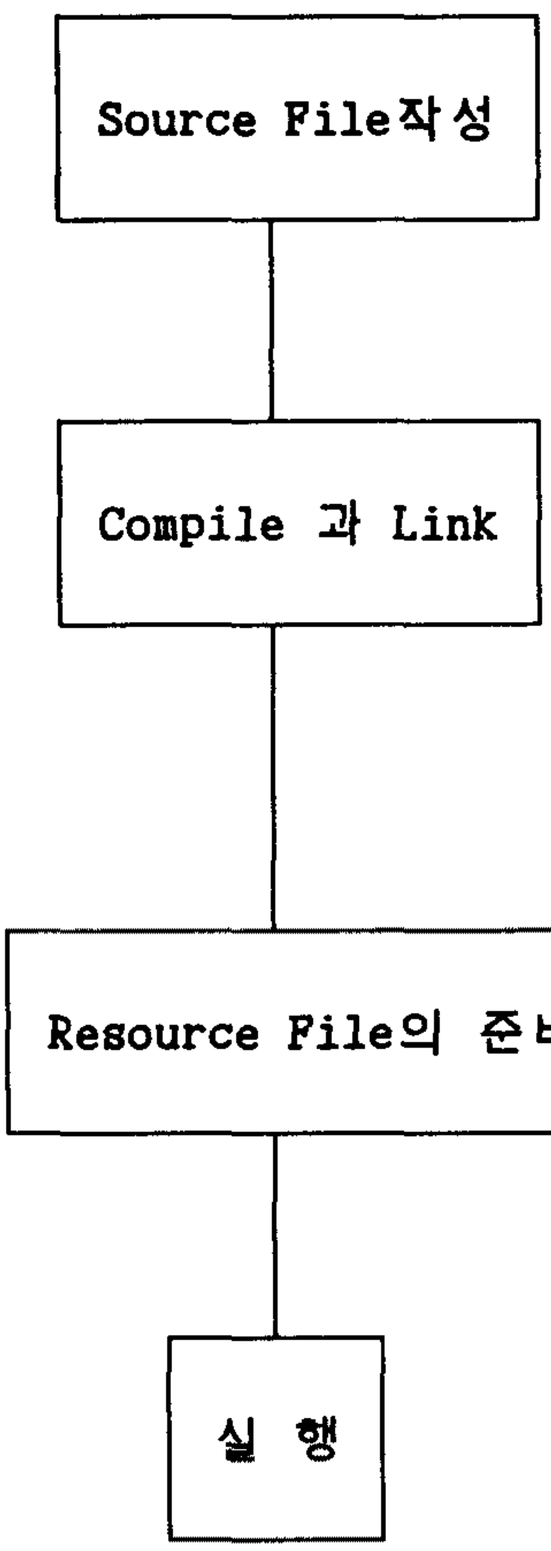
ToggleButtonGadget중의 오직 하나만이 set 상태를 갖도록 관리한다.

라. Naming Conventions

대구분	소구분	규약	예
Xlib	Xlib 함수	X로 시작하며 이하 각 단어의 첫 글자는 대문자	XCreateWindow() XOpenDisplay()
	Xlib 마크로	첫 글자는 X가 아닌 대문자로 시작하며 이하 각 단어의 첫 글자는 대문자	DisplayWidth() DefaultScreen()
Xt Intr-insics	함수 및 마크로	Xt로 시작하며 이하 각 단어의 첫 자는 대문자	XtCreateWidget() XtSetArg()
	Resource Name	XtN 으로 시작	#define XtNwidth "width"
	Resource Class	XtC 로 시작	#define XtCBackgr-ound"Background"
	Resource Representation	XtR 로 시작	#define XtRInt "Int"
Motif Widget set	Motif 함수	Xm으로 시작	XmCreatePushButton XmTextGetString()
	Resource Name	XmN으로 시작	#define XmNwidth "width"
	클래스명	Xm으로 시작하며 이하 각 단어의 첫 글자는 대문자	XmText XmPushButton
	클래스 포인터	xm으로 시작하며 이하 각 단어의 첫 글자는 대문자	xmRowColumnWidget-Class xmPushButtonWidge-tClass

[그림 A-7] Motif Naming Convention

마. 프로그램의 실행



예) #include <Xm/Xm.h>

#include <Xm/widget의 종류.h>

예) cc source_file.c -lXm -lXt -lXll

lXm : motif library

lXt : X toolkit library

lXll : Xlib의 library

예) Xdefaults File의 편집

XENVIRONMENT의 설정

/usr/lib/X11/app-defaults에 File준비

예) 필요한 X Toolkit Option 을 첨부한다.

예) cc -o sample sample.c -lXm -lXt -lX11 로 Compile 과 Link를 한 후 sample을 치면 원하는 작업이 수행된다.

XmfileselectionBox Widget은 libPW.a 라이브러리가 추가로 필요하다.

1) Utility Library의 생성

예) xs_concat_words()을 Utility Lib libXs에 저장

주프로그램: memo

library header file=> extern xmString xs_concat_words();

utility lib에 저장 => cc -c concat.c

ar ruv libXs.a concat.o

compile & link => cc -o memo memo.c -lXs -lXm -lXt -lX11

2) Header Files

/usr/include 디렉토리 하에 존재한다. 다음과 같이 특정 화일 일수록 나중에 기입한다.

예) # include <stdio.h>

include <X11/Xlib.h> /* Xlib header */

include <X11/Xutil.h> /* Xlib utility header */

include <X11/Intrinsic.h> /* Intrinsics header */

include <X11/StringDefs.h> /* 자원명 화일 */

include <Xm/Xm.h> /* Widget set header */

include <Xm/Label.h> /* XmLabel Widget header */

..... /* other Widget headers */

3) Default Files

프로그램용: /usr/lib/X11/app-defaults/<class>

사용자용: 사용자 home directory/. Xdefaults

예) /usr/lib/X11/app-defaults /* 모든 클래스에 적용 */

/usr/lib/X11/app-defaults/XMdemos /*XMdemos 클래스용 */

부록 : 참고 문헌

1) Douglas A. Young,『The X Window System programming and Applications with Xt』, OSF/Motif edition, Prentice-Hall,1990

2) 오기성,차경권,이인복,『UNIX에 기반을 둔 GUI를 지원하기위한 OPEN LOOK, OSF/Motif 적용방안에 관한 연구』, 한국 정보 과학회 가을 학술 논문집, vol.18, No2, 1991

3) Ralph R,Swick & Mark S.Ackerman, 『The X Toolkit : More Bricks for Building User-Interfaces』, winter, 1988 USENIX Conference, page (221 .. 233)

4) George A. Champine,『A Model for Distributed Campus Computing』, Digital Press, 1991

5) Edward Balkovich,Steven Lerman,Richard P. Parmelee, 『Computing in Higher Education: the Athena Experience』, Communication of the ACM, vol.28, No 11, November, 1985

6) 이 계영 편저,『X Window : Application Programming』, 지산사, 1991

7) 『X Window System Programming: Xlib』, Student Workbook, HP, 1989

8) Valerie Quercia & Tim O'Reilly, 『X Window System User's Guide』 ,OSF/Motif Edition, vol.3, O'Reilly & Associates, Inc. , 1991

9) Adrian Nye & Tim O'Reilly, 『X Toolkit Intrinstics Programming Manual』, OSF/Motif Edition,vol.4, O'Reilly & Associates, Inc. , 1990

10) 이 광형 & 이 재호, 『C++ 프로그래밍 언어』, 홍릉과학출판사, 1991

11) 권 용래,『Software Engineering』, Lecture Notes of KAIST , 1991

12) Jones, Oliver, 『Introduction to the X Window System』 ,Prentice-Hall,1989